Light and Shading

EECS 442 – Prof. David Fouhey

Winter 2019, University of Michigan

http://web.eecs.umich.edu/~fouhey/teaching/EECS442_W19/

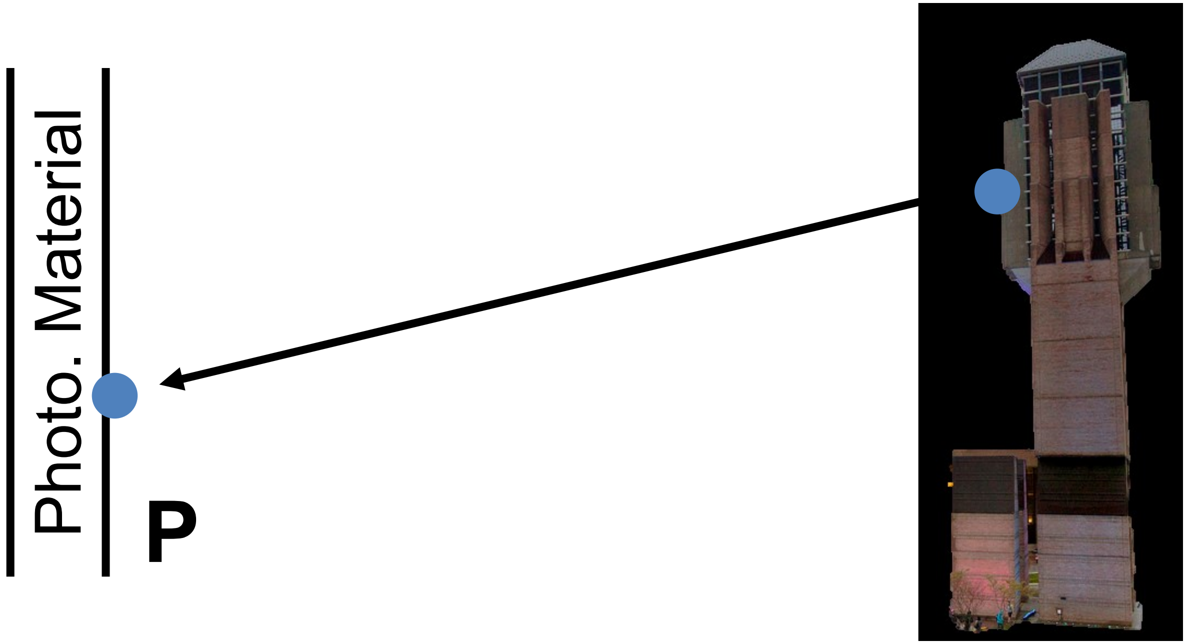
Administrivia

- I sent out requests for waitlist additions. I will continue to add as spots free up but chances are diminishing.
- ***PLEASE SIGN UP ON PIAZZA.*** There are no secrets on canvas.
- HW1 out. **Any general questions (not about content)?**
- Discussion on Wednesday: image processing / numpy. Materials out on piazza.

Recap: Projection

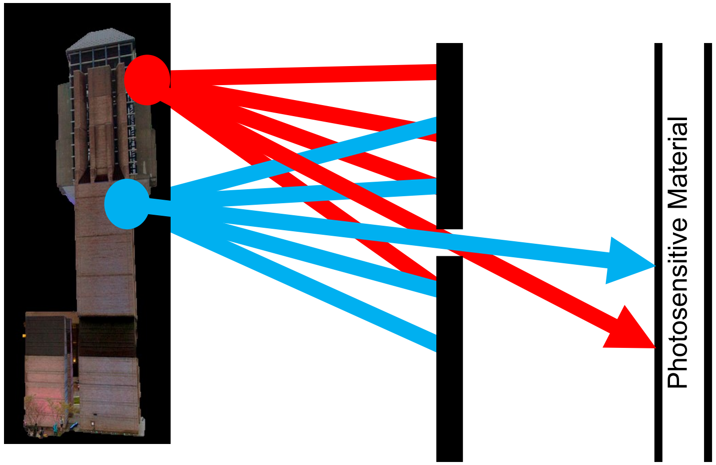
Image → $\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{X}$ ← *World*

Intrinsic ↗ ↖ *Extrinsic*



Recap: Lenses

Pinhole Model



Mathematically correct
Not quite correct in practice
Reasonable approximation

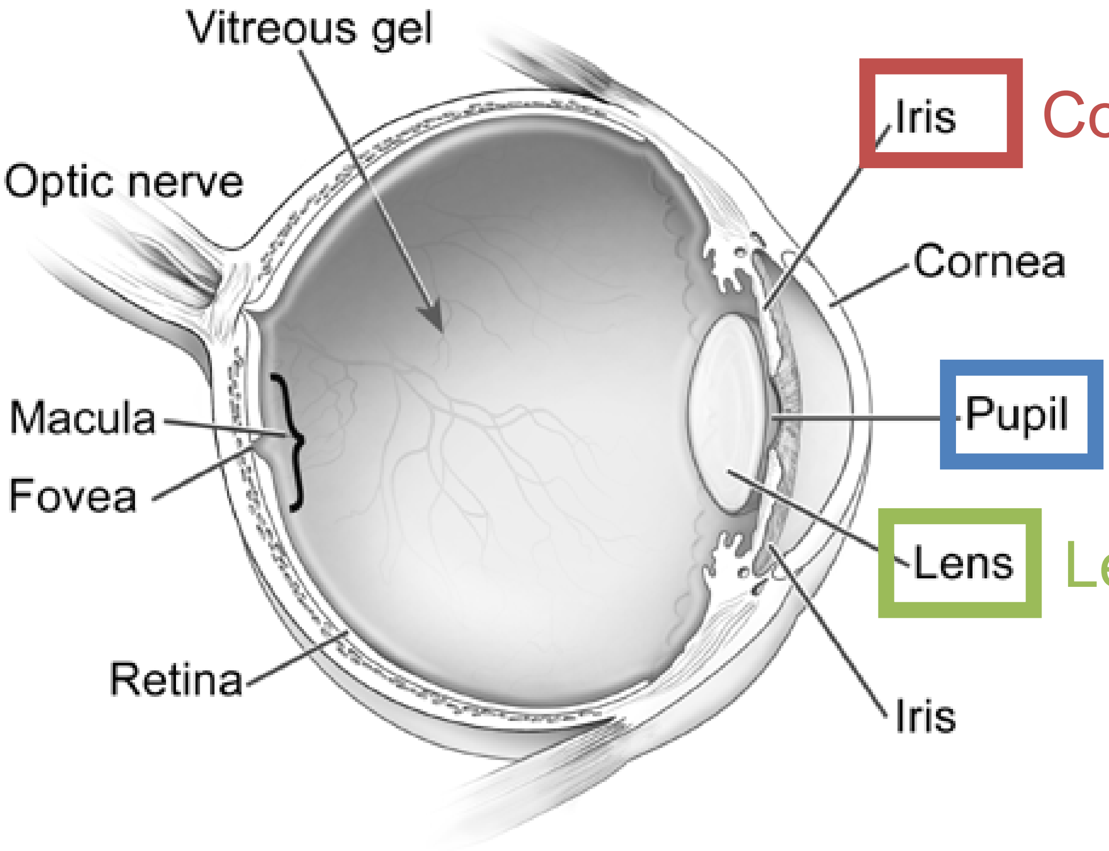
Reality: Lenses

Necessary in practice
Introduce complications
Complications fixable

Today

- A little bit about light and how you represent it
- A little bit about lighting and how it works

Your Very Own Camera



Controls pupil

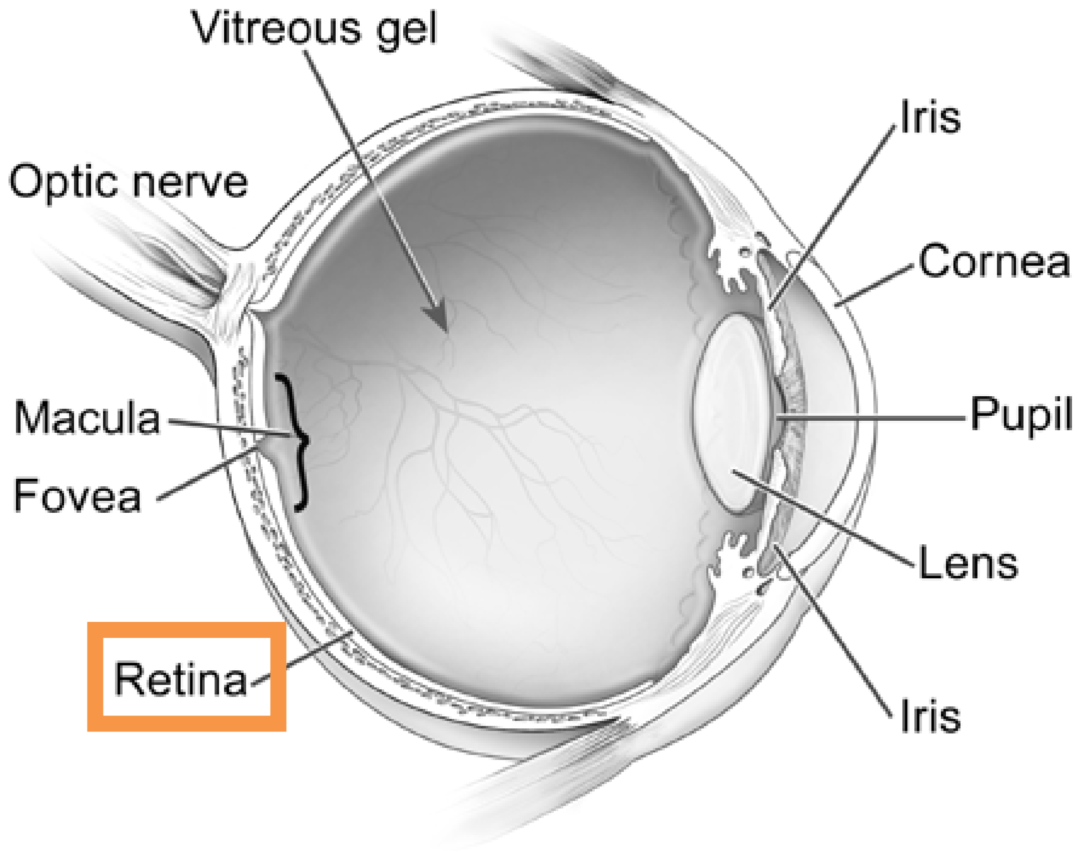
Aperture letting in light

Lens…

Where’s the film/CCD?

Slide Credit: NIH

Your Very Own Camera

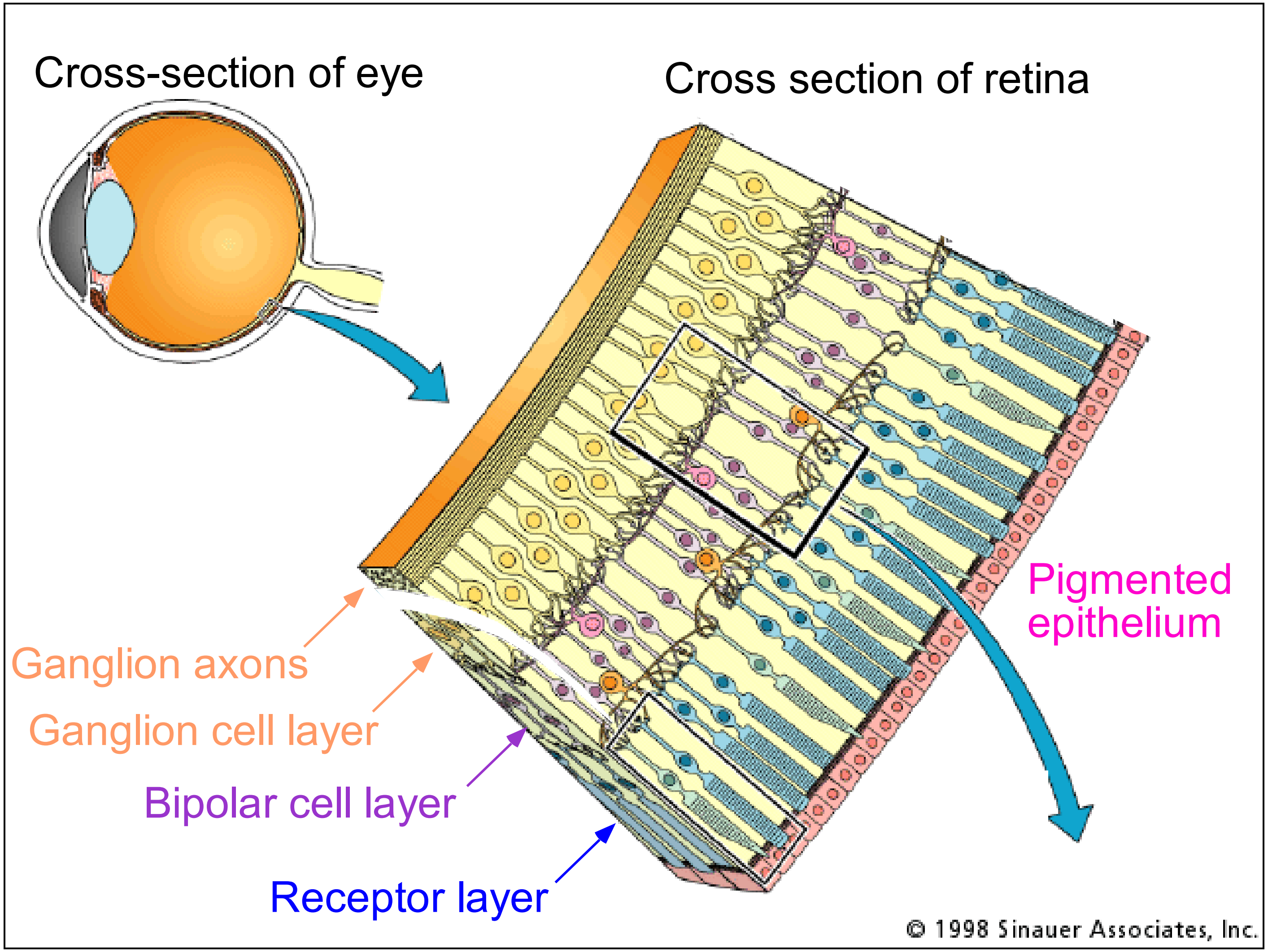


Where's the film/CCD?

Slide Credit: NIH

Demo Time

What is Retina/Film Made Of?



Cross-section of eye

Cross section of retina

Ganglion axons

Ganglion cell layer

Bipolar cell layer

Receptor layer

Pigmented epithelium

Slide Credit: J. Hays

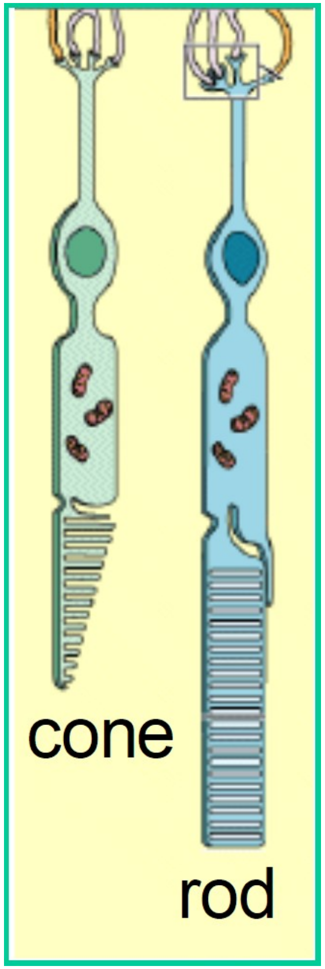
Two Type of Photo Receptors

Cones

- cone-shaped
- less sensitive
- operate in high light
- color vision

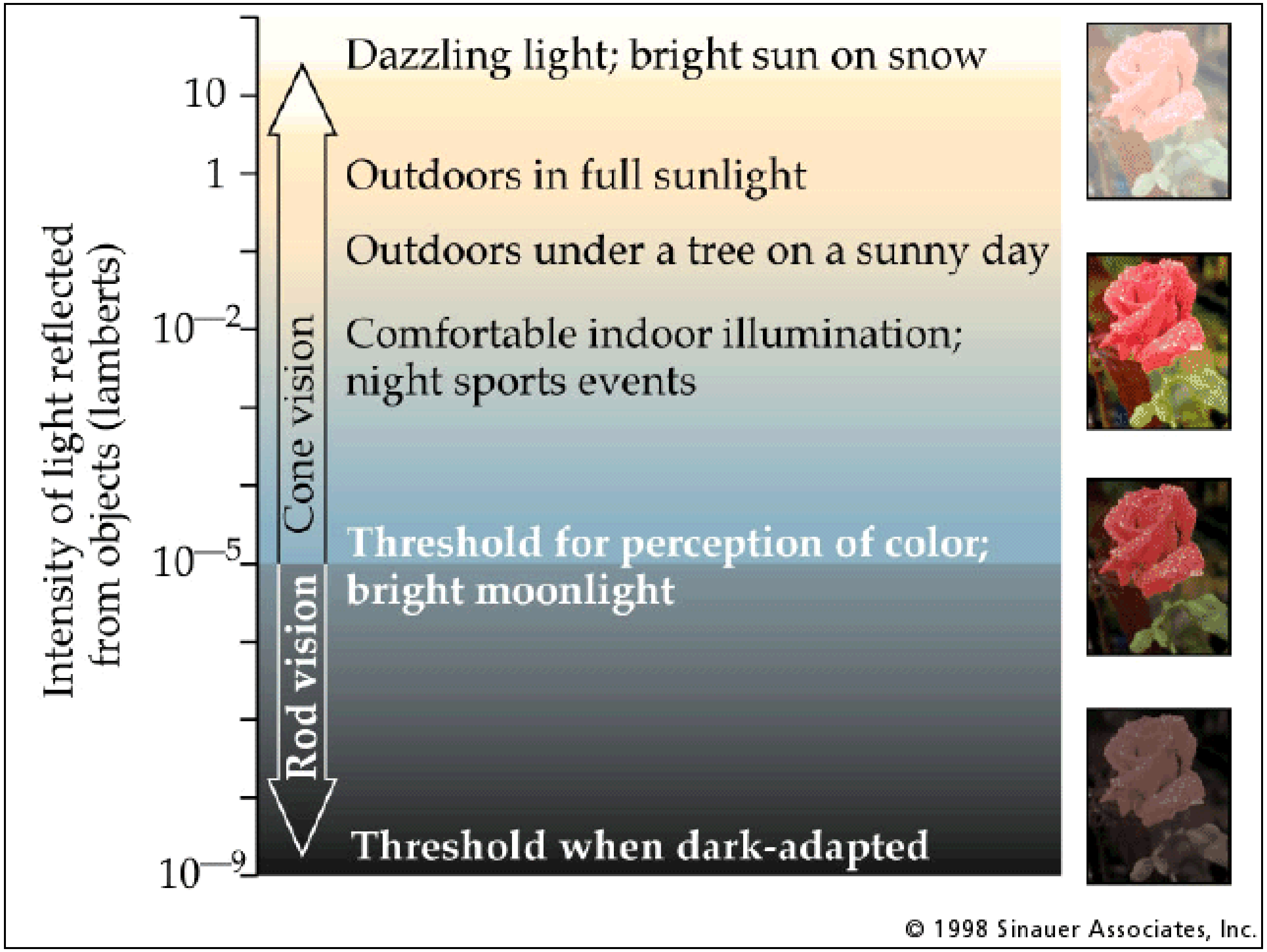
Rods

- rod-shaped
- highly sensitive
- operate at night
- gray-scale vision



Slide Credit: J. Hays

Rod / Cone Sensitivity



Slide Credit: J. Hays

Rod/Cone Distribution

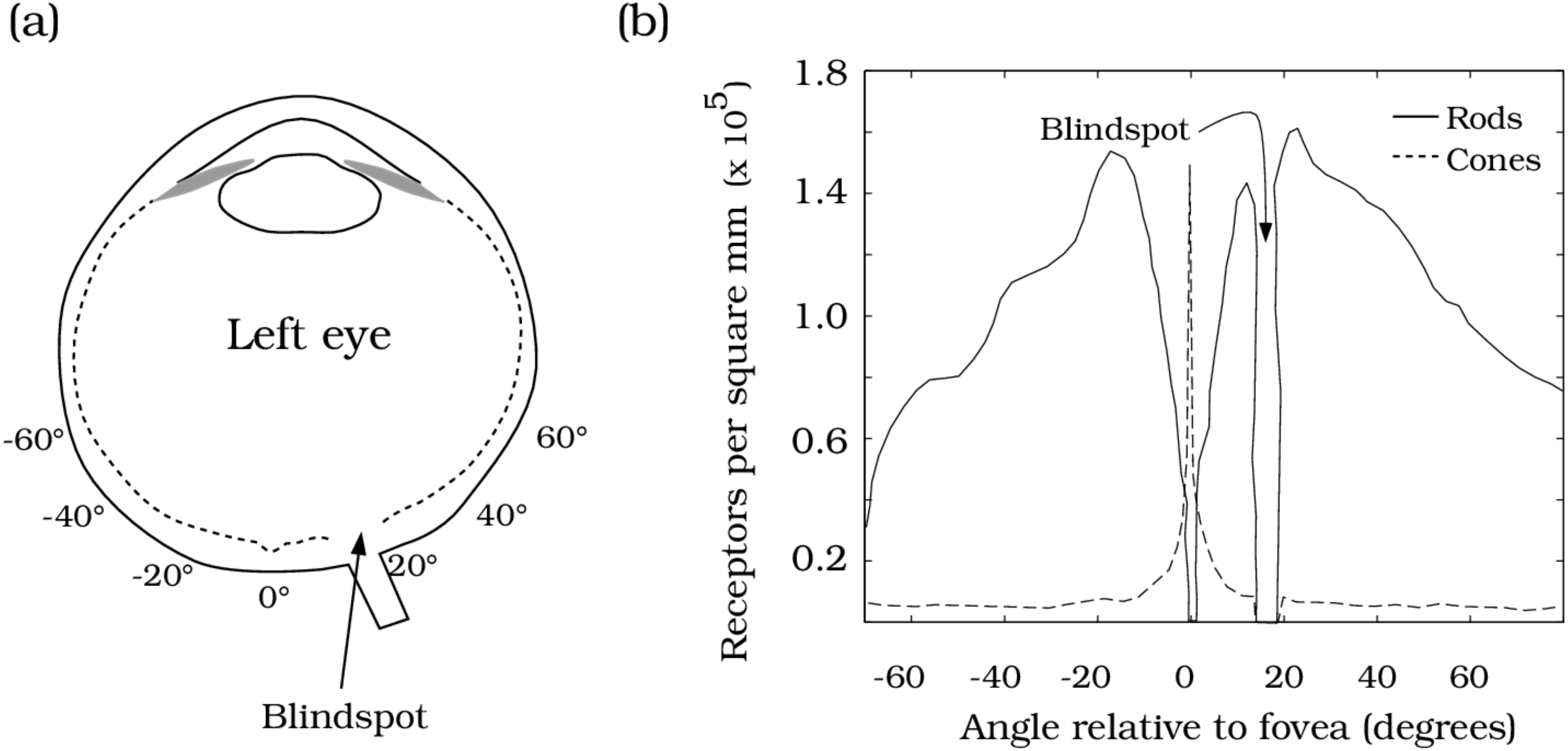
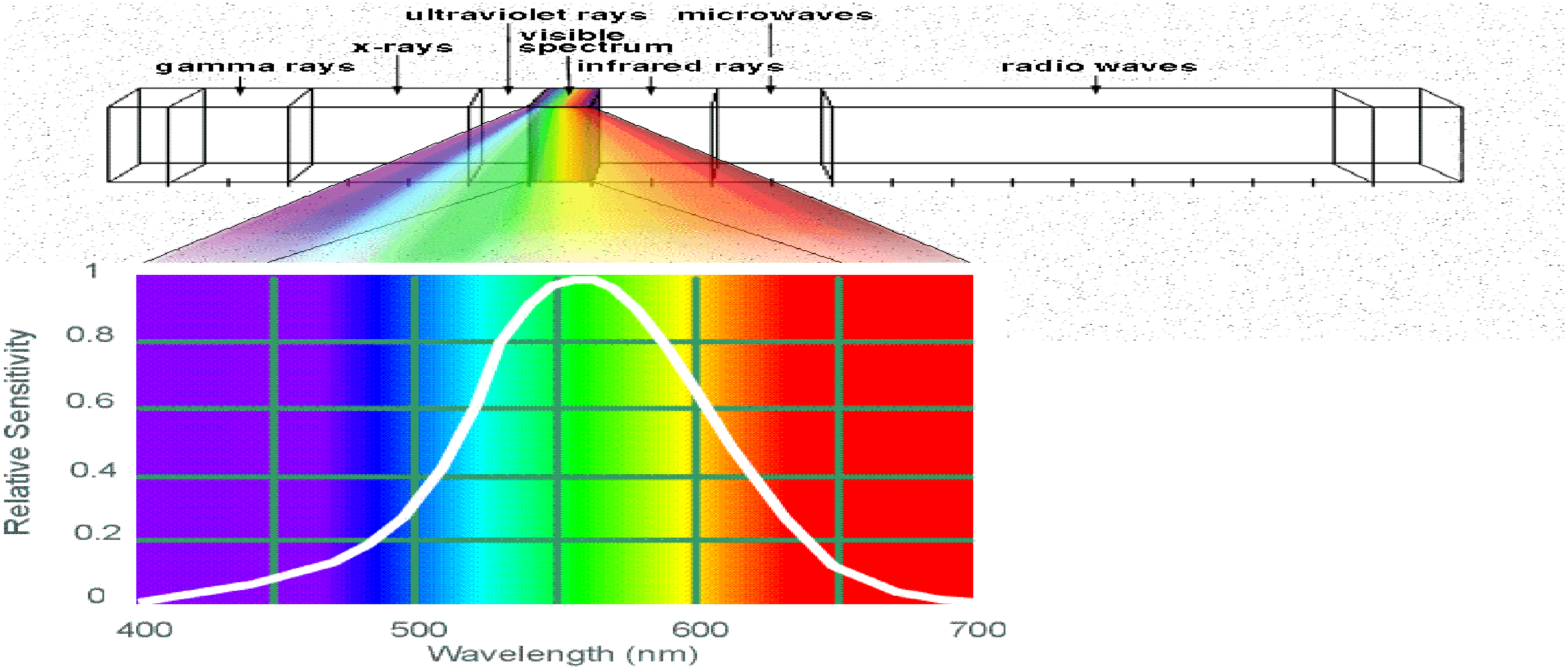


Diagram Credit: B. A. Wandell, *Foundations of Vision*

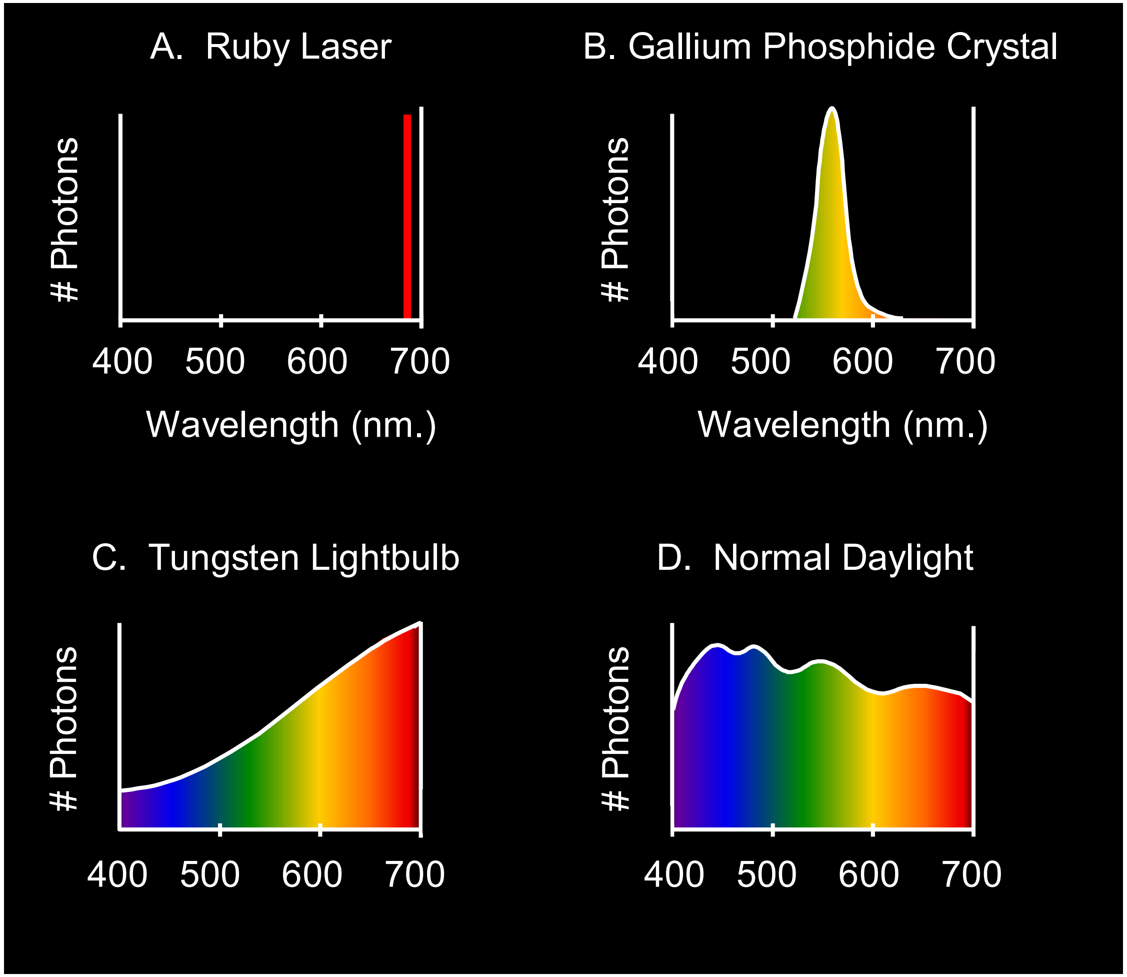
Electromagnetic Spectrum



Why do we see light in these wavelengths?

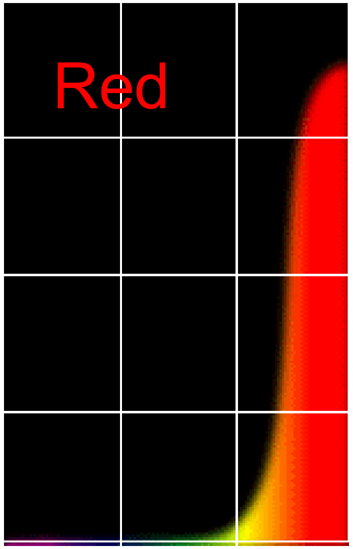
Slide Credit: J. Hays

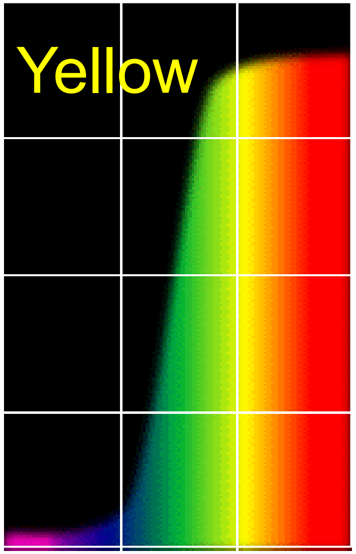
The Physics of Light

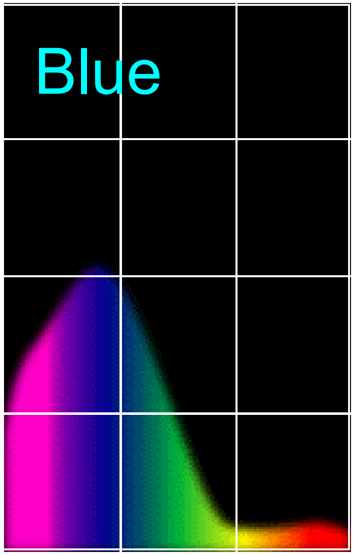


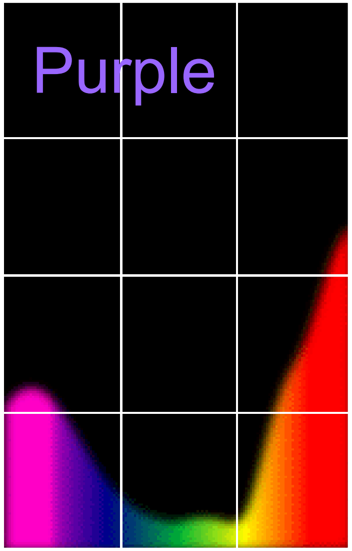
Slide Credit and Copyright: S. Palmer

The Physics of Light



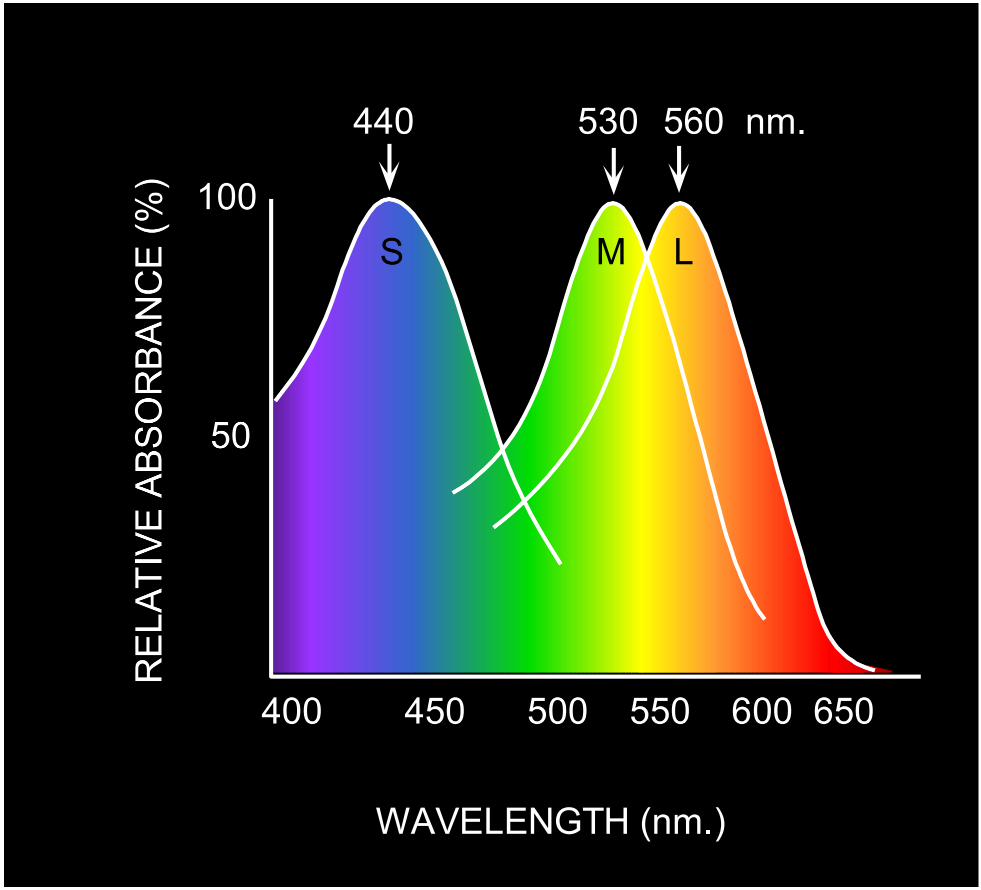




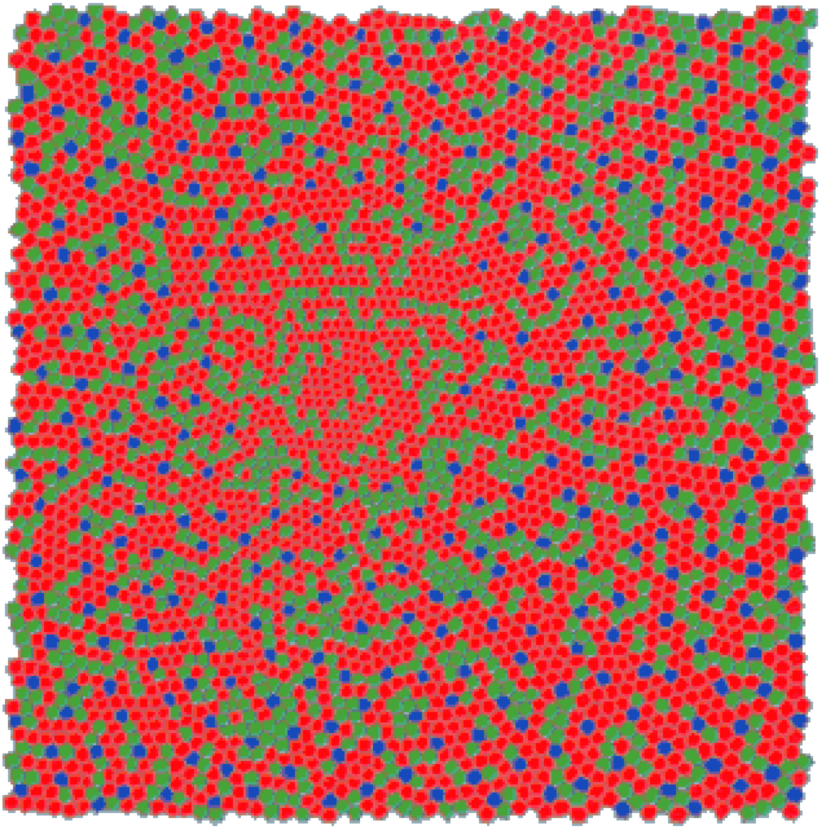


Slide Credit and Copyright: S. Palmer

The Physics of Light



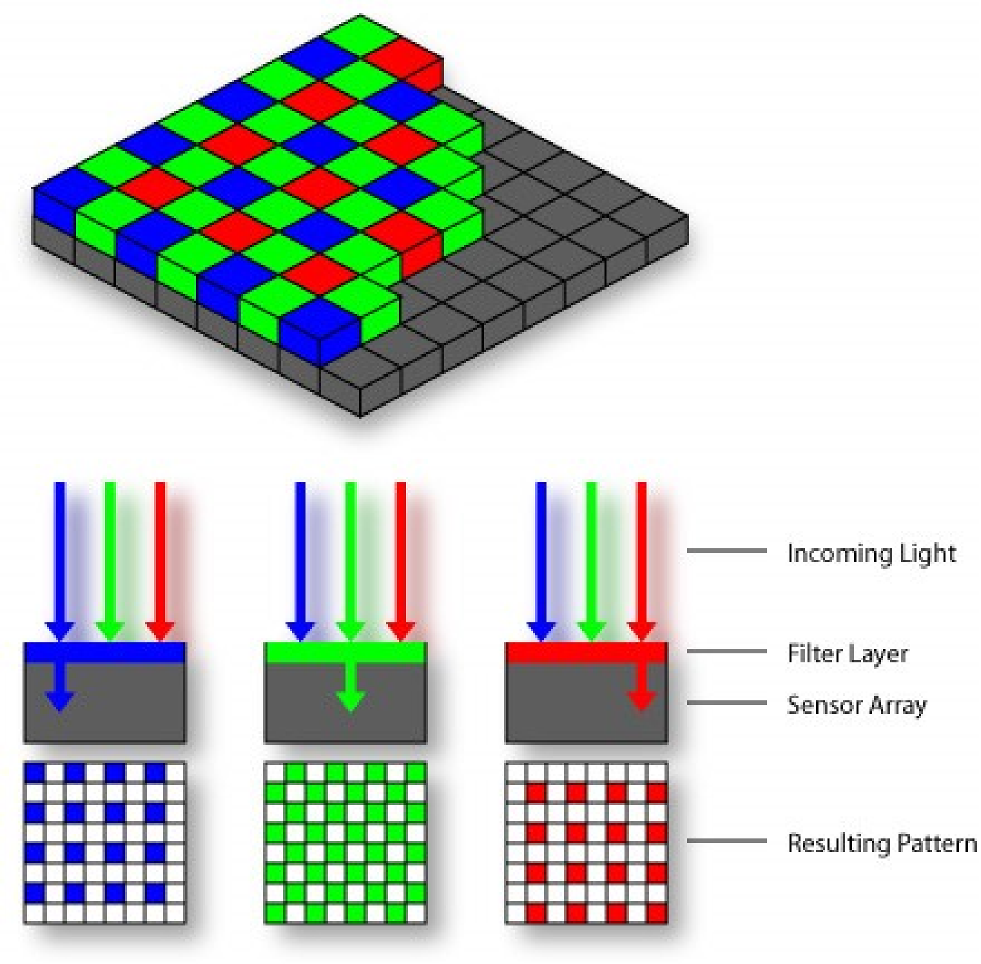
Cone mosaic

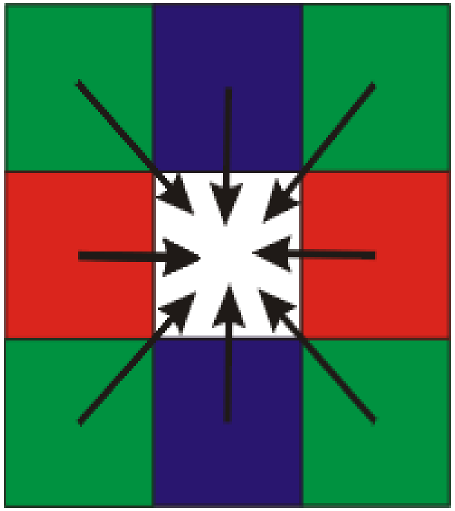


Slide Credit and Copyright: S. Palmer

How Do We Get Light?

Artificial Cones



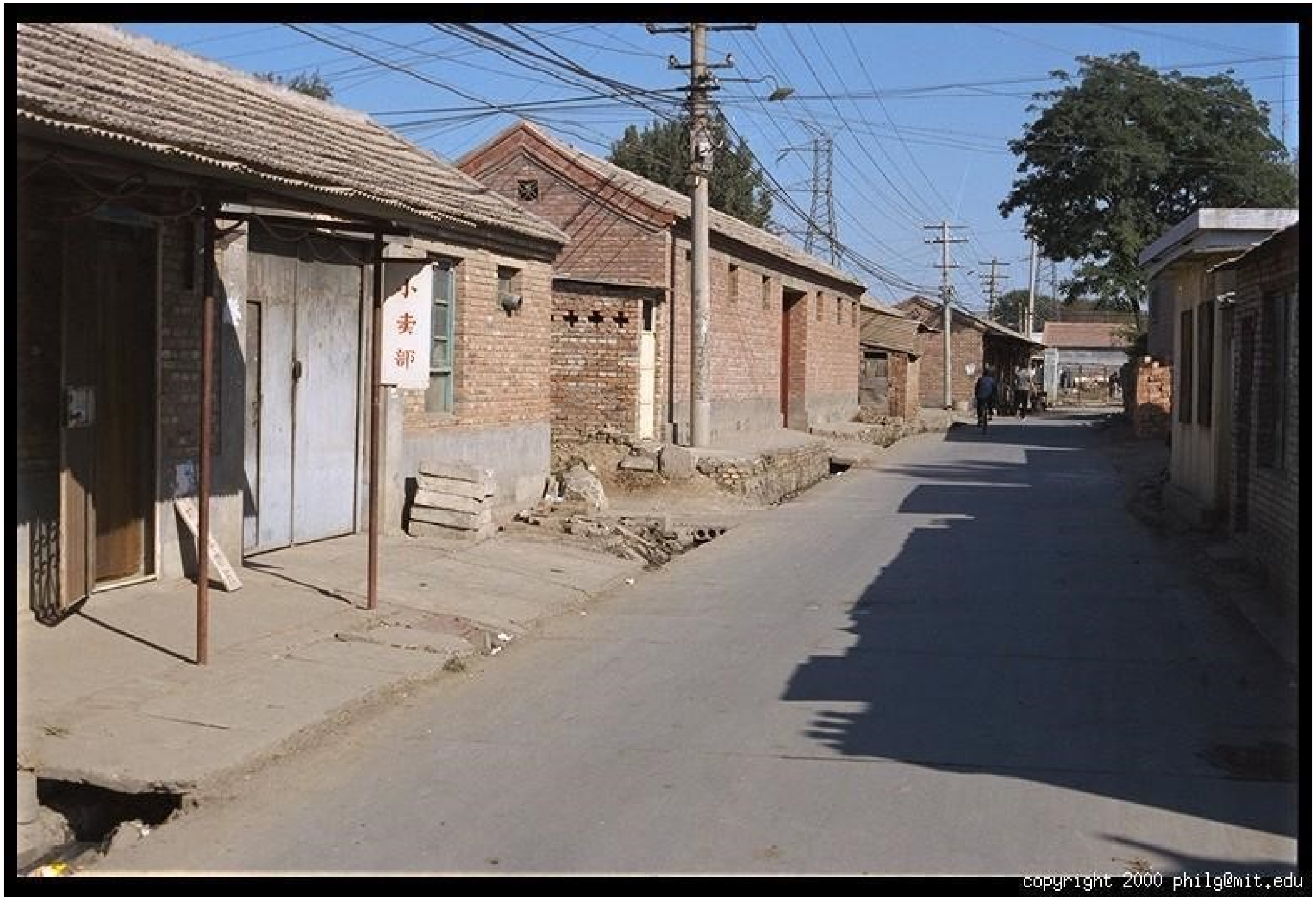


Estimate RGB at ‘G’ cells from neighboring values

Slide Credit: S. Seitz

Slide by Steve Seitz

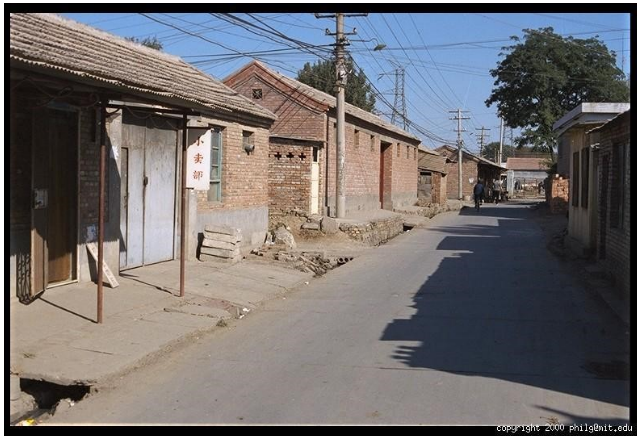
Color Image



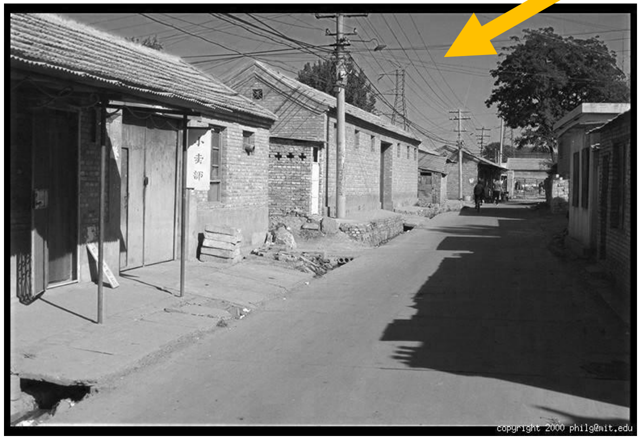
Slide Credit: J. Hays

Color Image

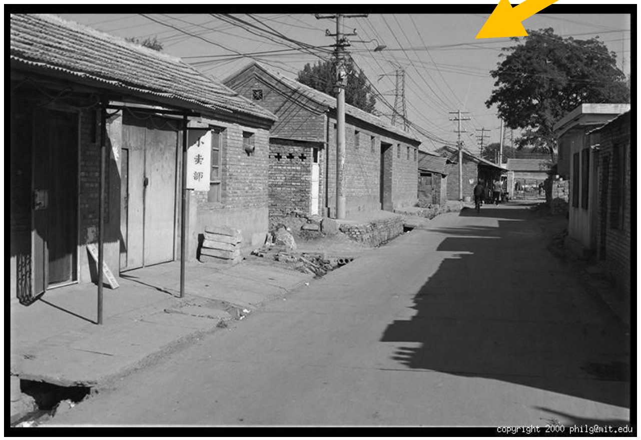
Combined



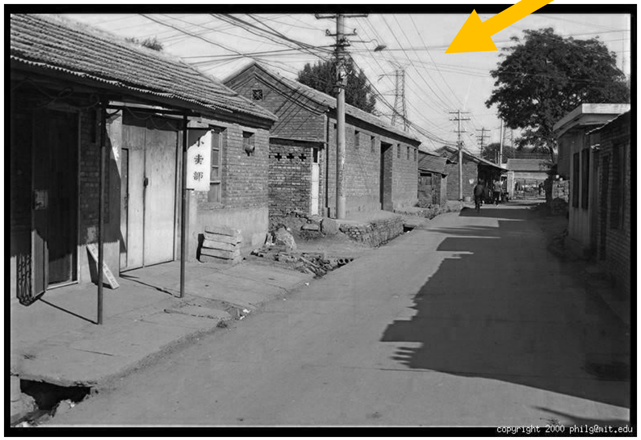
Red



Green

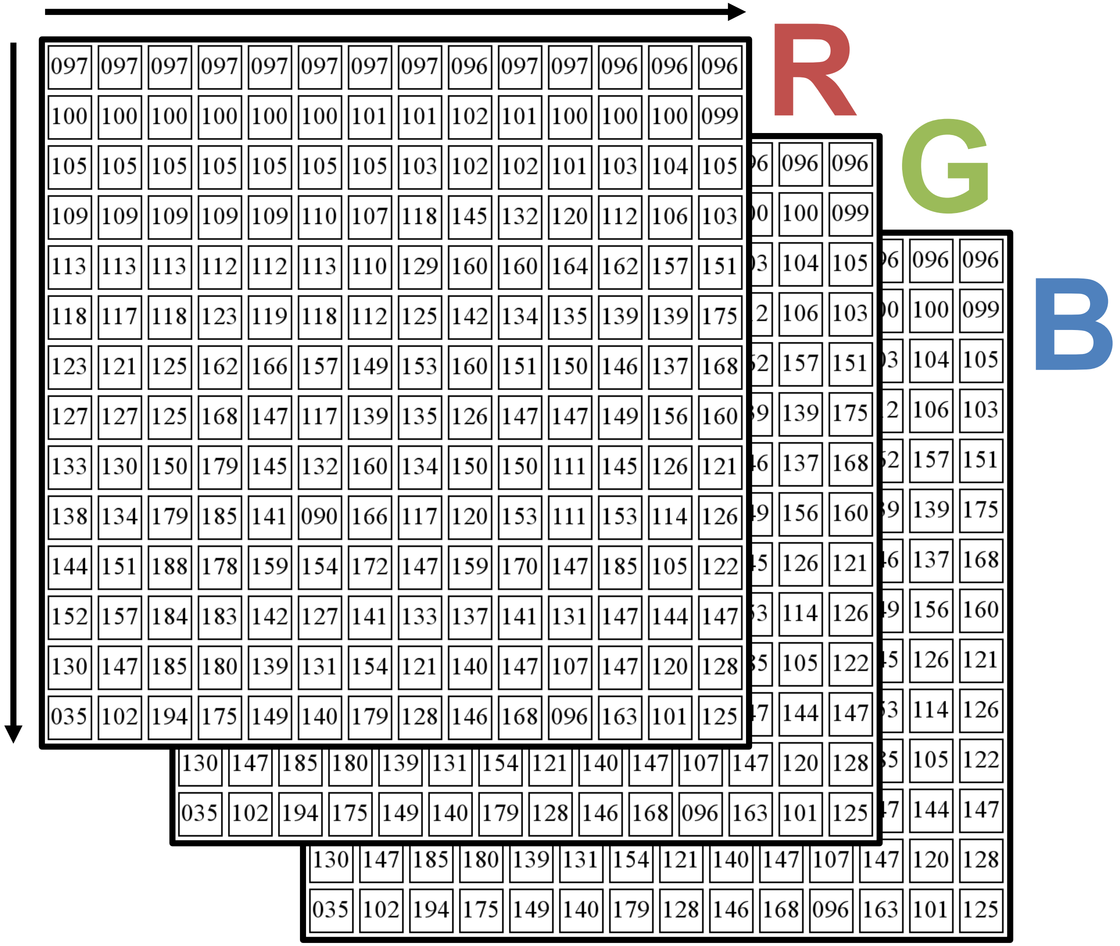


Blue



Slide Credit: J. Hays

Images in Python



097	097	097	097	097	097	097	097	096	097	097	096	096	096
100	100	100	100	100	100	101	101	102	101	100	100	100	099
105	105	105	105	105	105	105	103	102	102	101	103	104	105
109	109	109	109	109	110	107	118	145	132	120	112	106	103
113	113	113	112	112	113	110	129	160	160	164	162	157	151
118	117	118	123	119	118	112	125	142	134	135	139	139	175
123	121	125	162	166	157	149	153	160	151	150	146	137	168
127	127	125	168	147	117	139	135	126	147	147	149	156	160
133	130	150	179	145	132	160	134	150	150	111	145	126	121
138	134	179	185	141	090	166	117	120	153	111	153	114	126
144	151	188	178	159	154	172	147	159	170	147	185	105	122
152	157	184	183	142	127	141	133	137	141	131	147	144	147
130	147	185	180	139	131	154	121	140	147	107	147	120	128
035	102	194	175	149	140	179	128	146	168	096	163	101	125

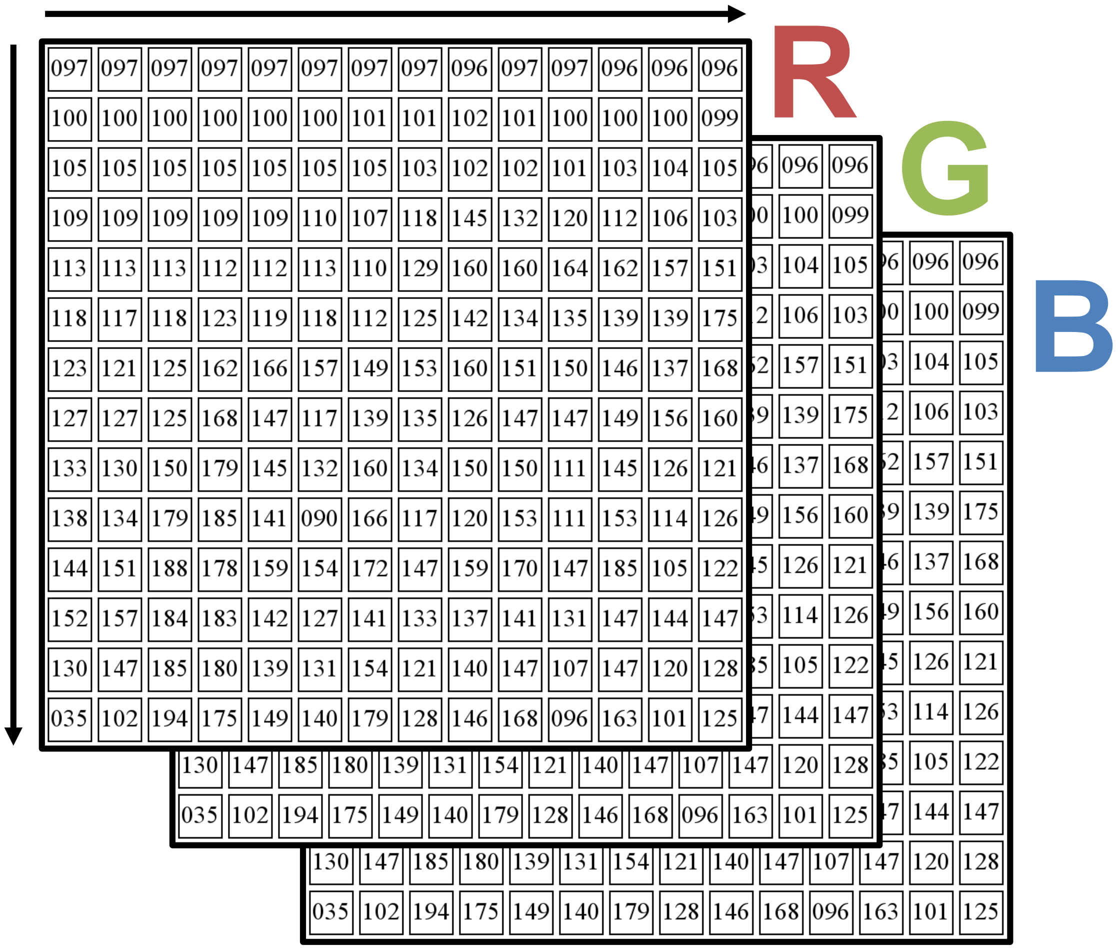
Images in Python

Images are matrix / tensor `im`

`im[0,0,0]`
top, left, red

`im[y,x,c]`
row y, column x, channel c

`im[H-1,W-1,2]`
bottom right blue



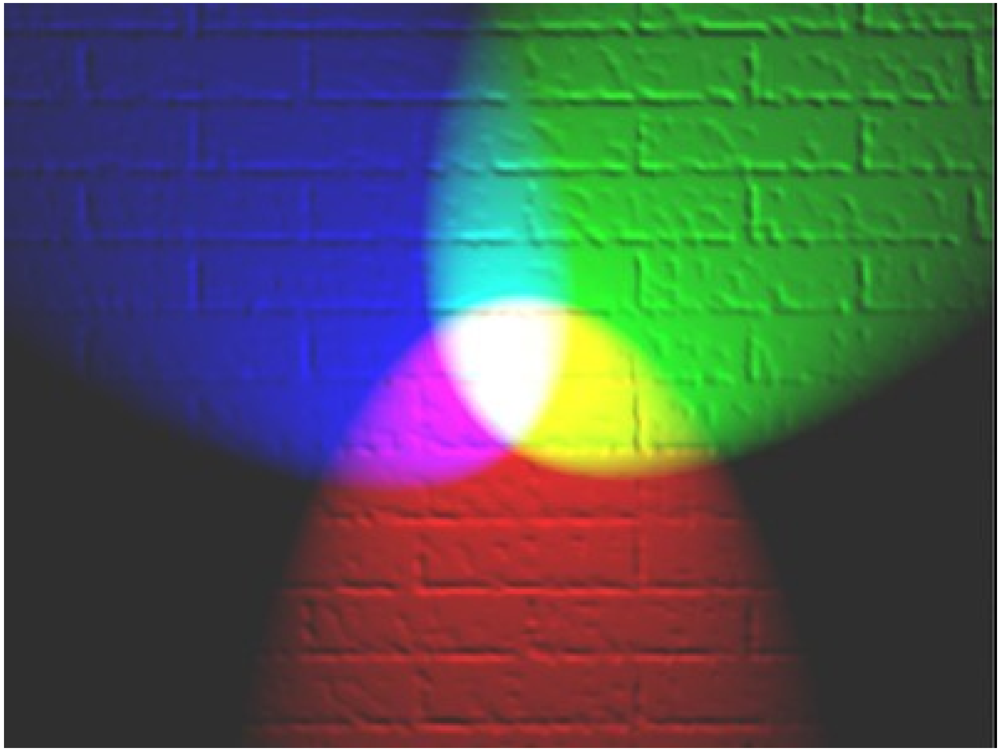
097	097	097	097	097	097	097	097	096	097	097	096	096	096
100	100	100	100	100	100	101	101	102	101	100	100	100	099
105	105	105	105	105	105	105	103	102	102	101	103	104	105
109	109	109	109	109	110	107	118	145	132	120	112	106	103
113	113	113	112	112	113	110	129	160	160	164	162	157	151
118	117	118	123	119	118	112	125	142	134	135	139	139	175
123	121	125	162	166	157	149	153	160	151	150	146	137	168
127	127	125	168	147	117	139	135	126	147	147	149	156	160
133	130	150	179	145	132	160	134	150	150	111	145	126	121
138	134	179	185	141	090	166	117	120	153	111	153	114	126
144	151	188	178	159	154	172	147	159	170	147	185	105	122
152	157	184	183	142	127	141	133	137	141	131	147	144	147
130	147	185	180	139	131	154	121	140	147	107	147	120	128
035	102	194	175	149	140	179	128	146	168	096	163	101	125

Slide inspired by James Hays

5 Things To Always Remember

1. Origin is top left
2. Rows are first index (**what’s the fastest direction for accessing?)**
3. Usually referred to as Height x Width
4. Typically stored as uint8 [0,255]
5. for y in range(H): for x in range(W): will run 1 million times for a 1000x1000 image. *A 4GHz processor can do only 4K clock cycles per pixel per second.*

Representing Colored Light



Discussion time: how many numbers do you actually need for colored light? Assume all tuples (R,G,B) are legitimate colors (they are).

Image Credit: http://en.wikipedia.org/wiki/File:RGB_illumination.jpg

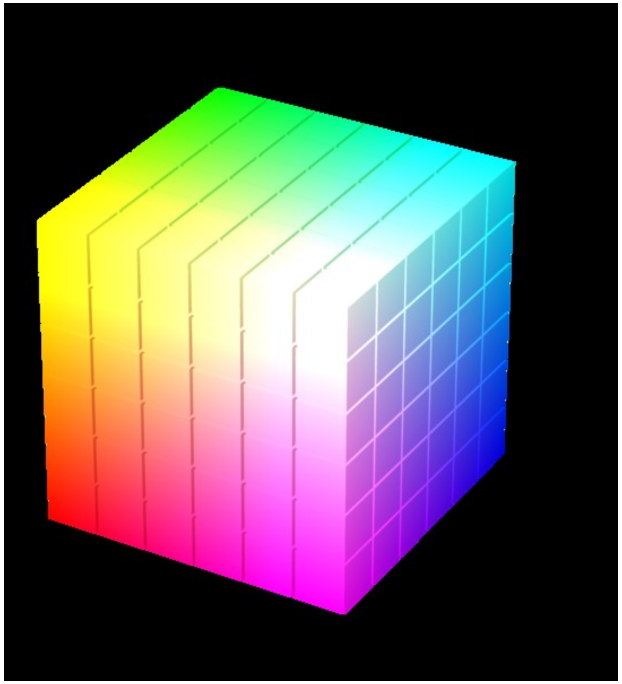
One Option: RGB

Pros

1. Simple
2. Common

Cons

1. Distances don't make sense
2. Correlated



R

G

B

Slide Credit: J. Hays, RGB cube: https://en.wikipedia.org/wiki/RGB_color_model

RGB

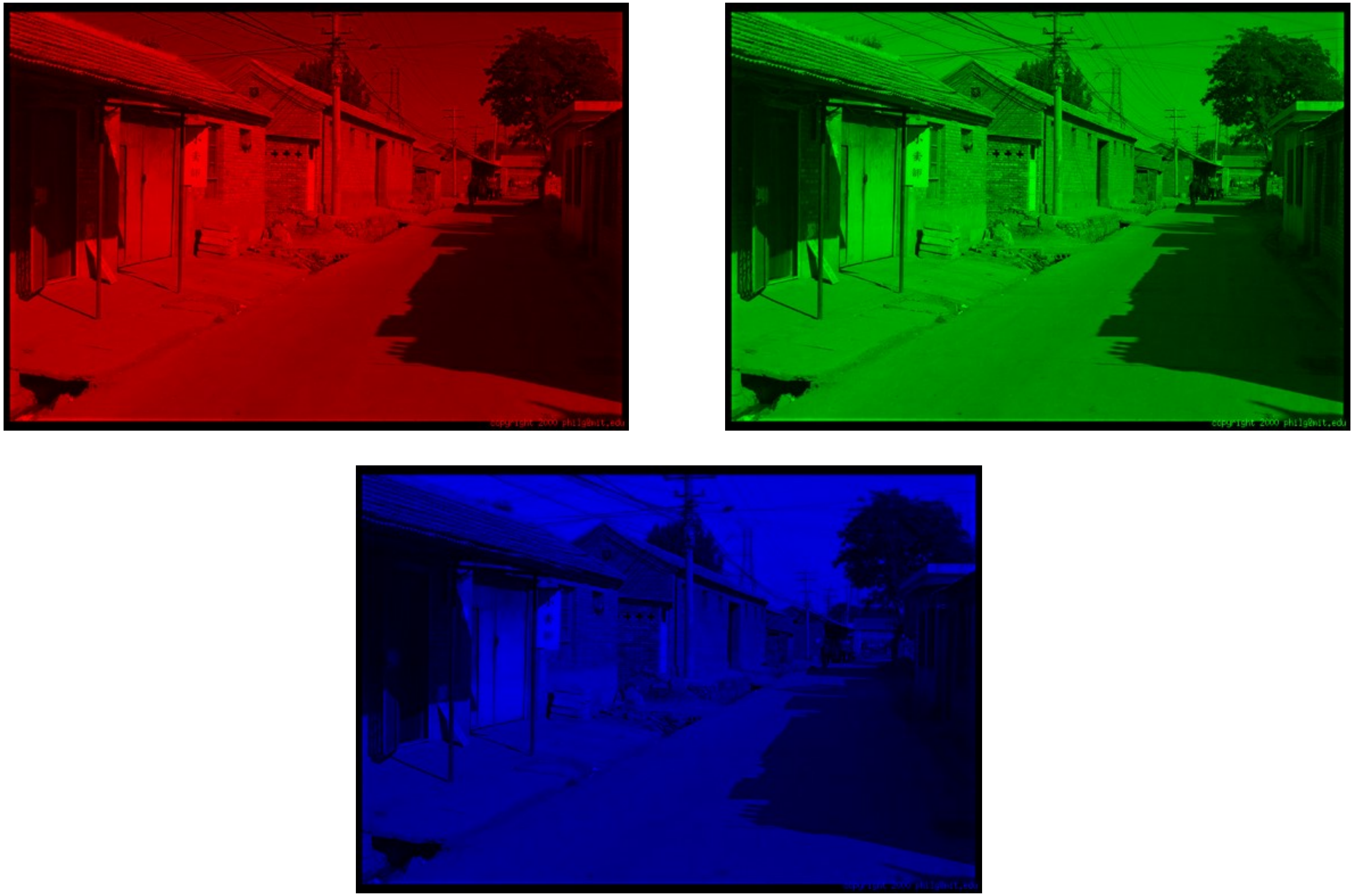

copyright 2000 philg@mit.edu
copyright 2000 philg@mit.edu
copyright 2000 philg@mit.edu

Photo credit: J. Hays

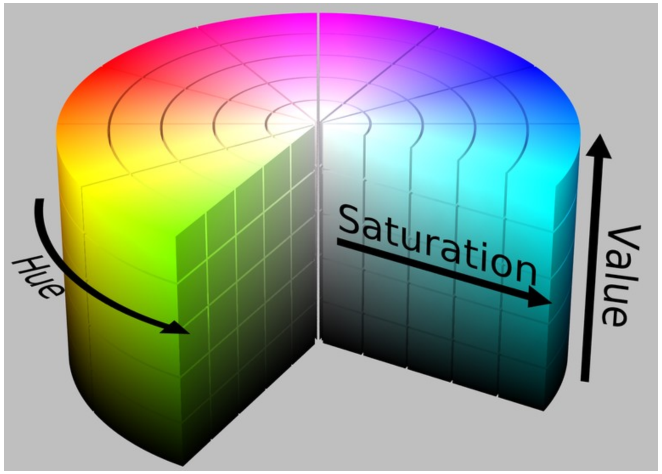
Another Option: HSV

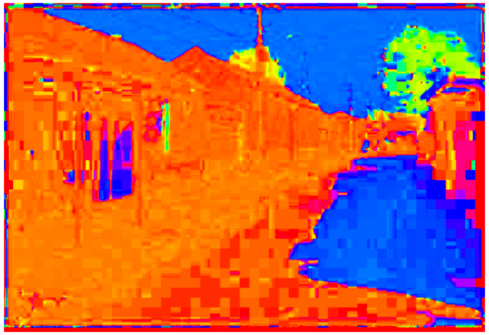
Pros

1. Intuitive for picking colors
2. Sort of common
3. Fast to convert

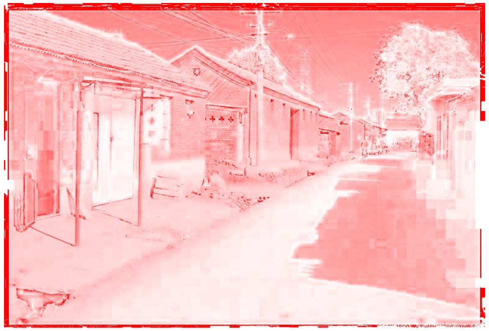
Cons

1. Not as good as other better spaces

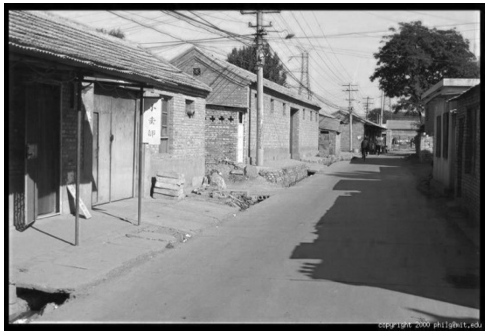




H
(S=1,V=1)



S
(H=1,V=1)



V
(H=1,S=0)

Slide Credit: J. Hays, HSV cylinder: https://en.wikipedia.org/wiki/HSL_and_HSV

HSV

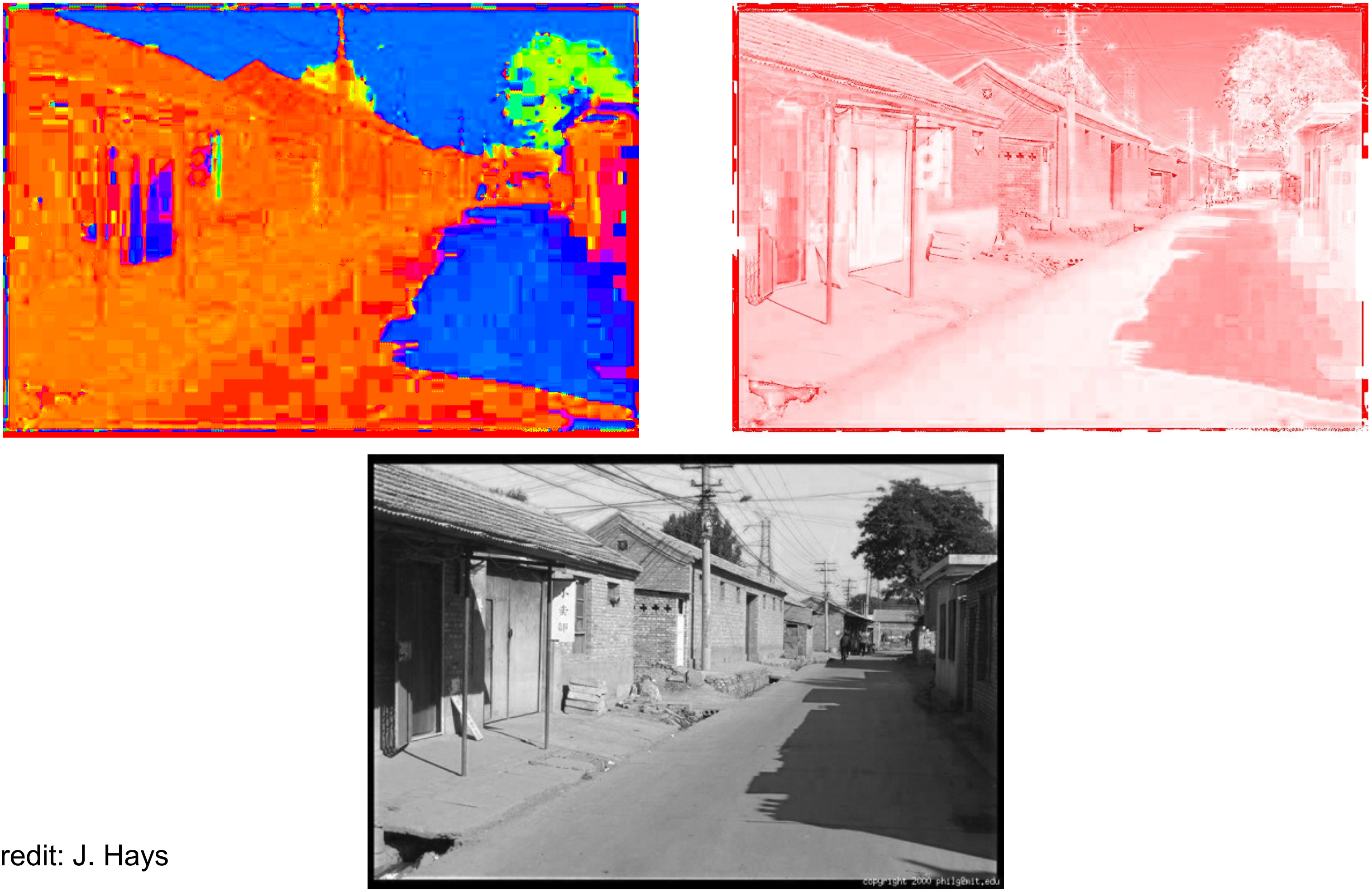


Photo credit: J. Hays

Another Option: YCbCr/YUV

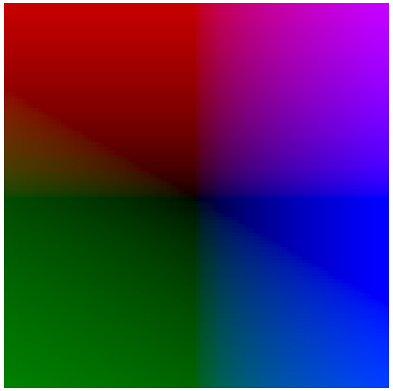
Pros

1. Great for transmission / compression

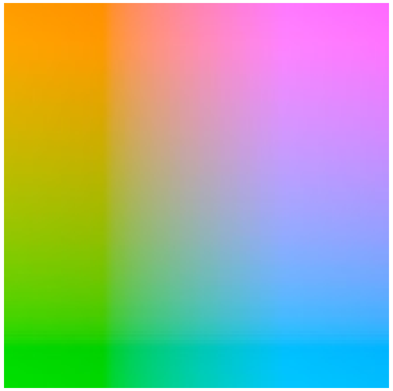
Cons

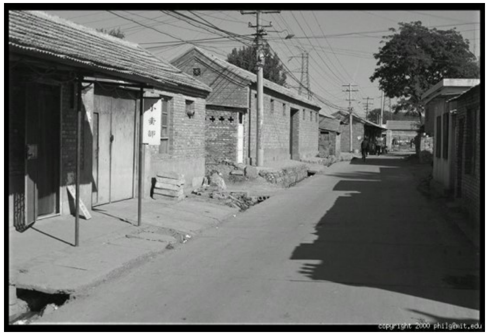
1. Not as good as other better smart color spaces

Y = 0



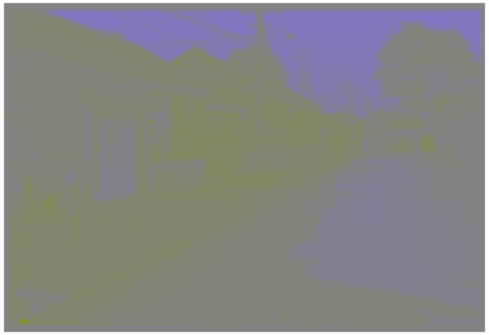
Y = 0.5





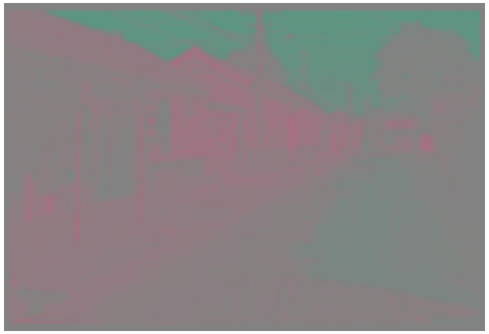
Y

(Cb=0.5, Cr=0.5)



Cb

(Y=0.5, Cr=0.5)

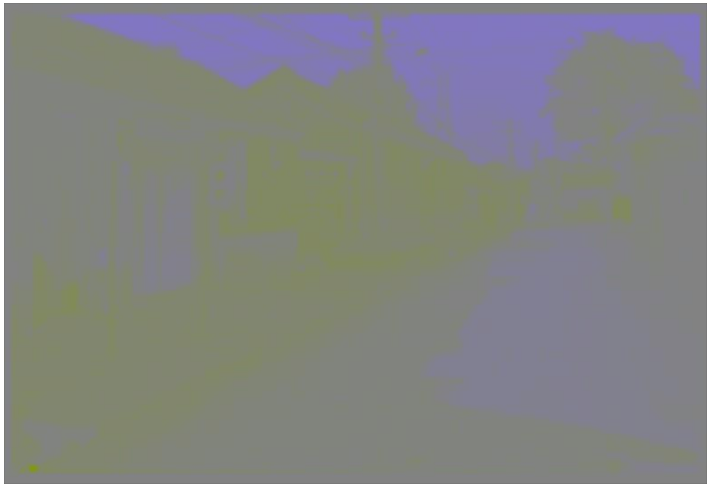


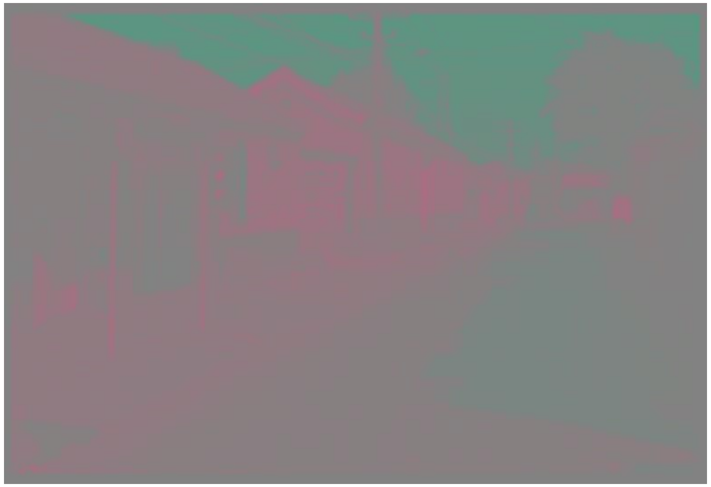
Cr

(Y=0.5, Cb=05)

Slide Credit: J. Hays, YUV cube: https://en.wikipedia.org/wiki/YUV

YCbCr





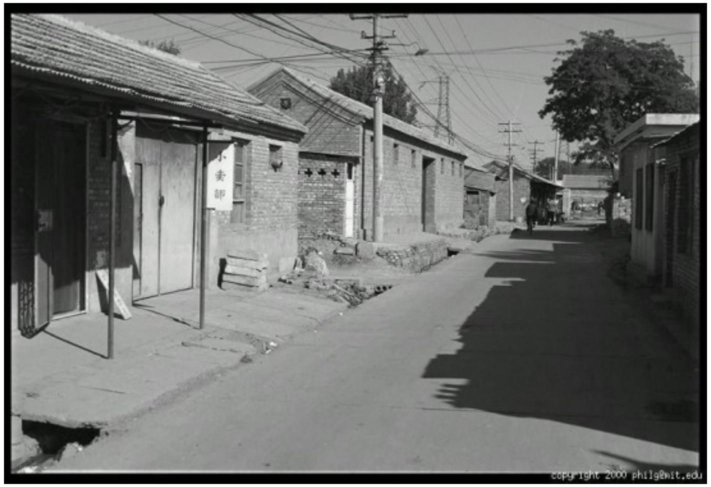


Photo credit: J. Hays

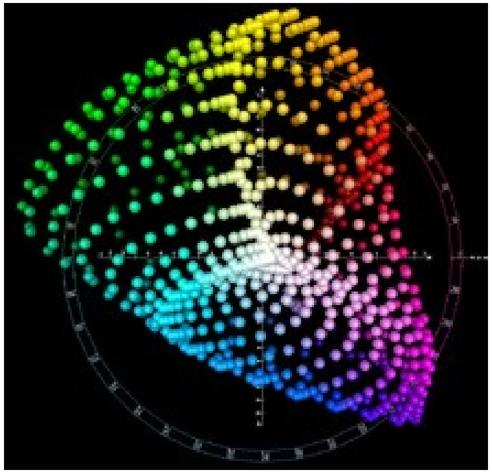
Another Option: Lab

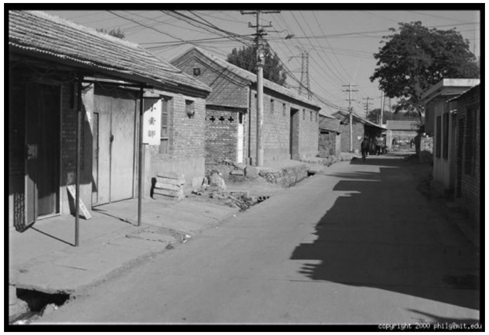
Pros

1. Distances correspond with human judgment
2. Safe

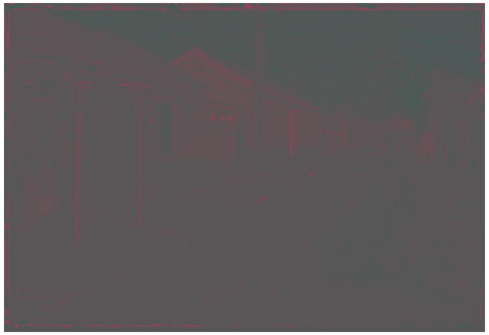
Cons

1. Complex to calculate (don't write it yourself, lots of fp calculations)

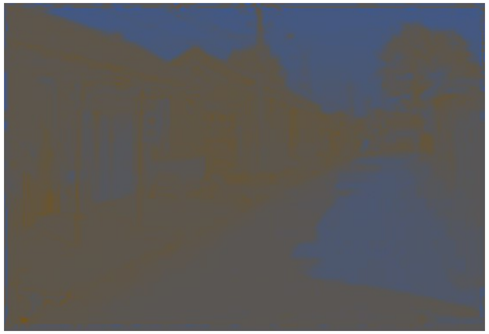




L
(a=0,b=0)



a
(L=65,b=0)



b
(L=65,a=0)

Slide Credit: J. Hays, Lab diagram cube: https://en.wikipedia.org/wiki/CIELAB_color_space

Lab

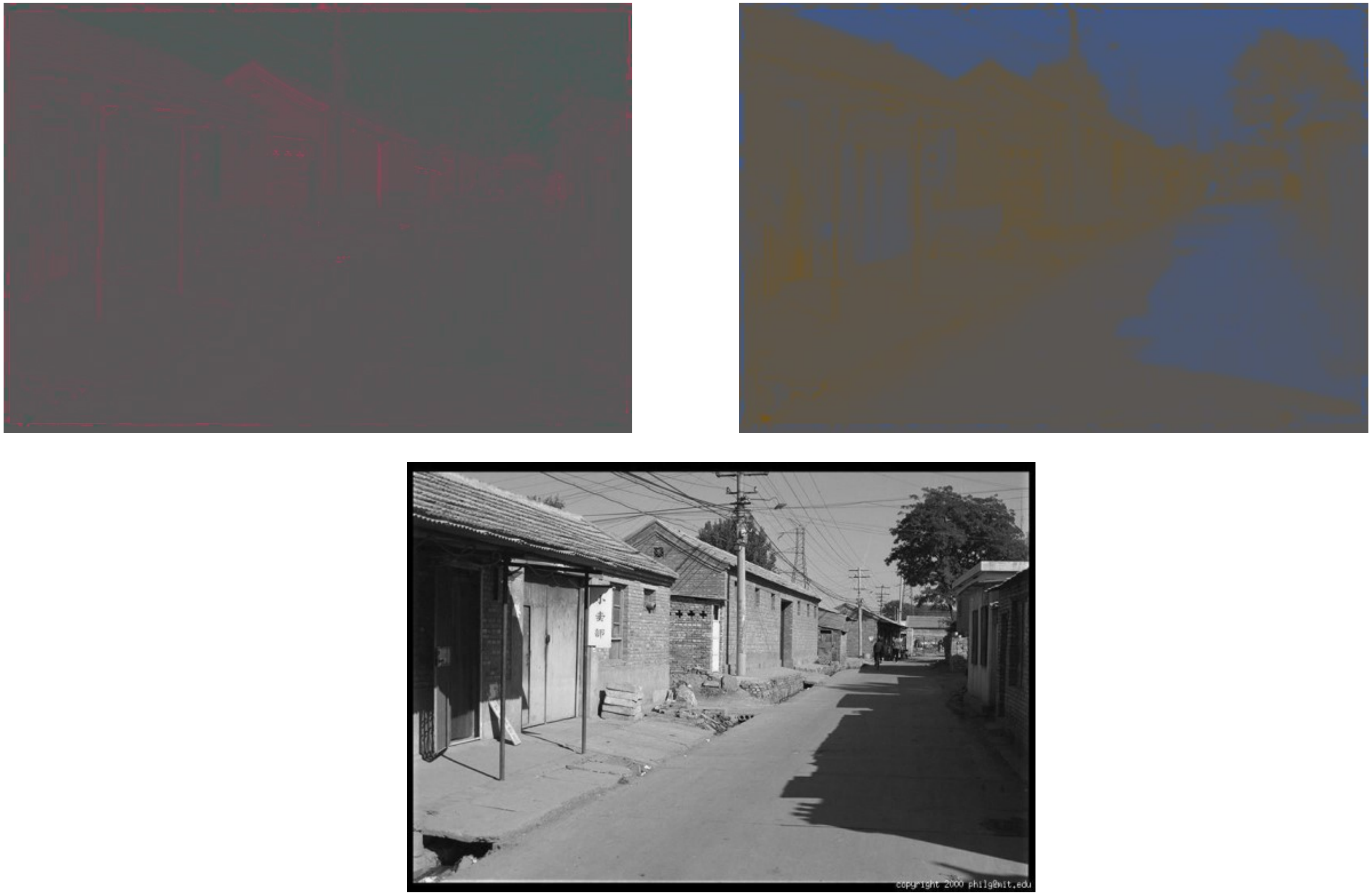


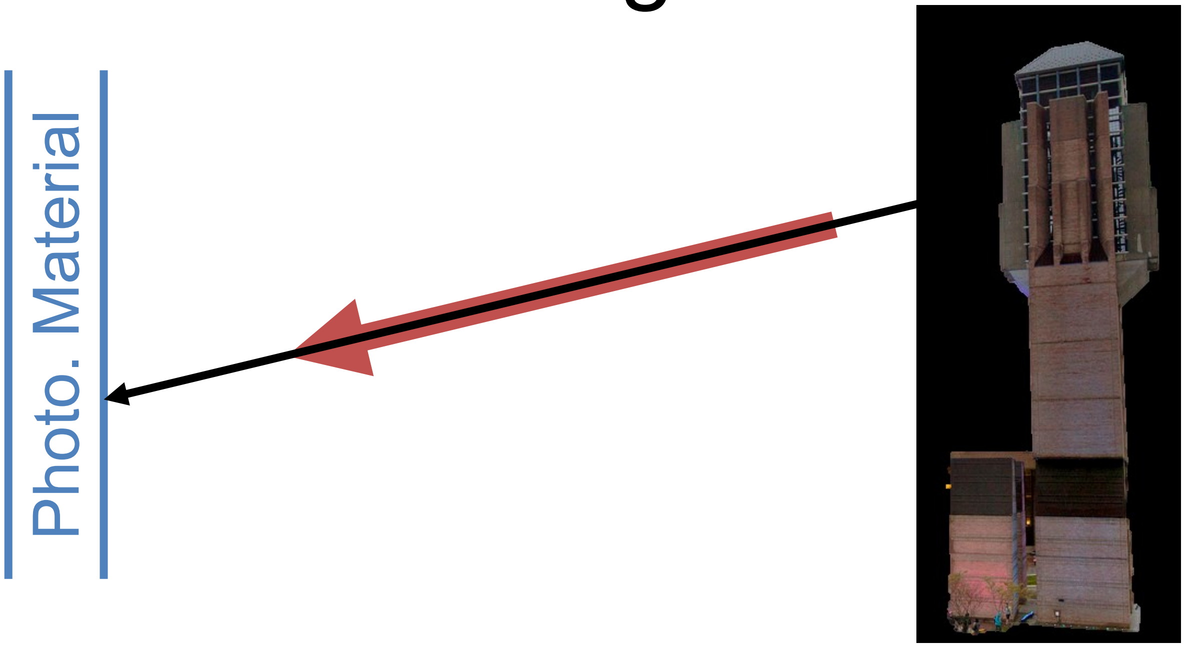
Photo credit: J. Hays

Why Are There So Many?

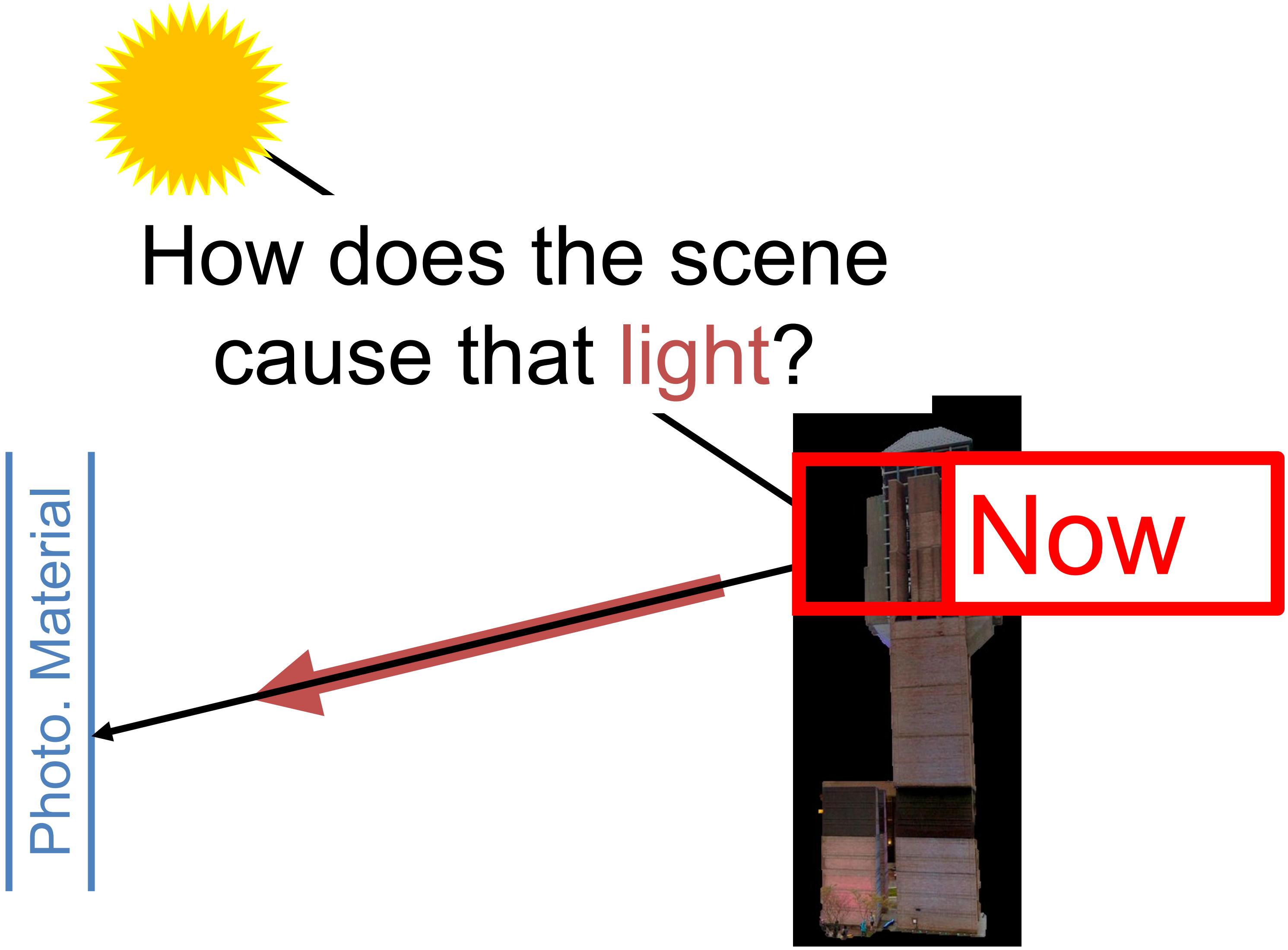
- Each serves different functions
  - RGB: sort of intuitive, standard, everywhere
  - HSV: good for picking, fast to compute
  - YCbCr/YUV: fast to compute
  - Lab: the right(?) thing to do, but “slow” to compute
- Pick based on what you need and don’t sweat it: color really isn’t crucial

So Far

How do we represent light
and its storage on film?



Now

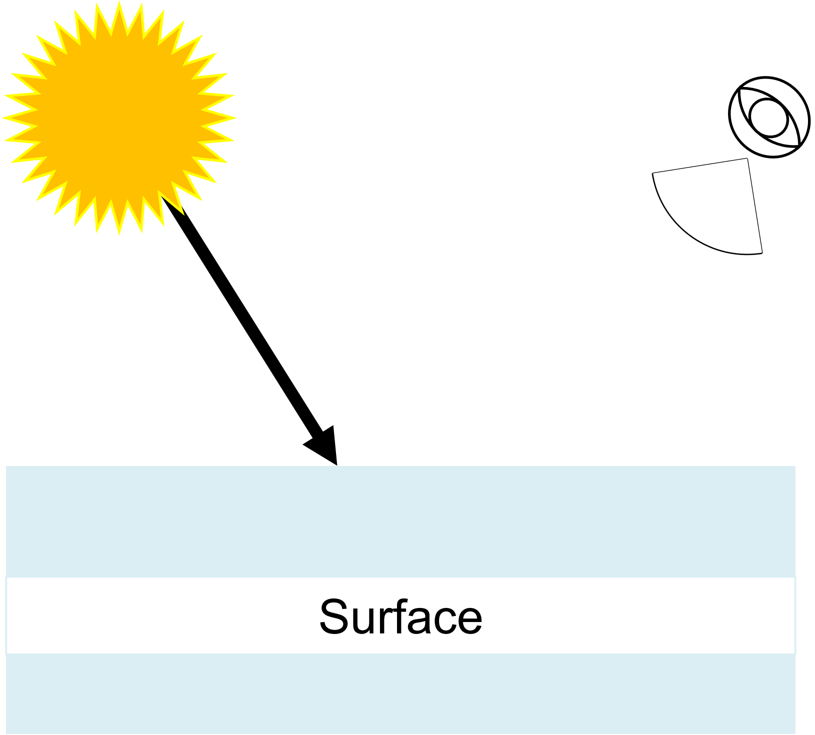


How does the scene cause that light?

Now

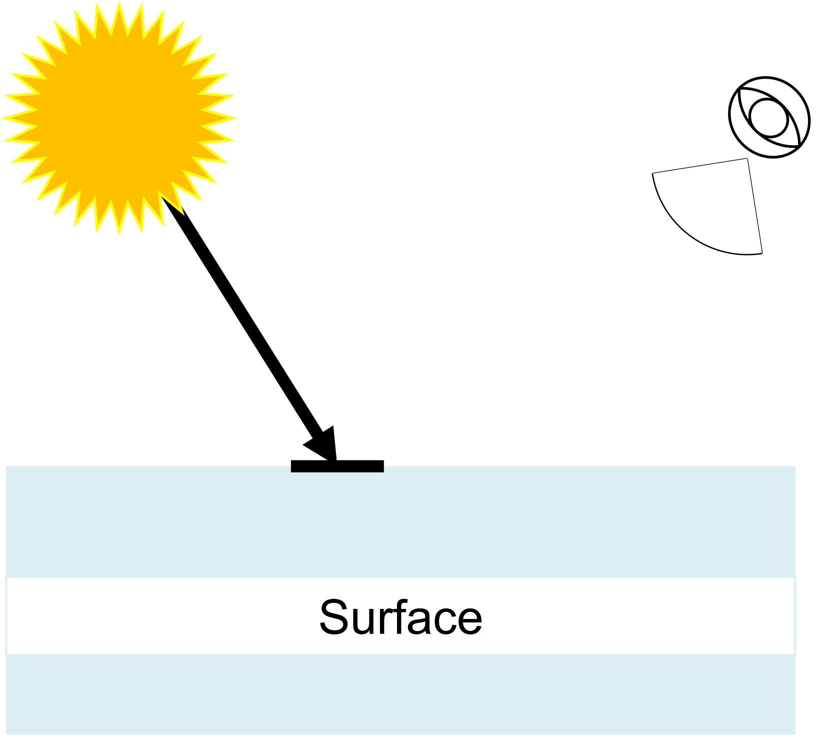
Photo. Material

Light and Surfaces



What happens when light hits a surface?

Light and Surfaces

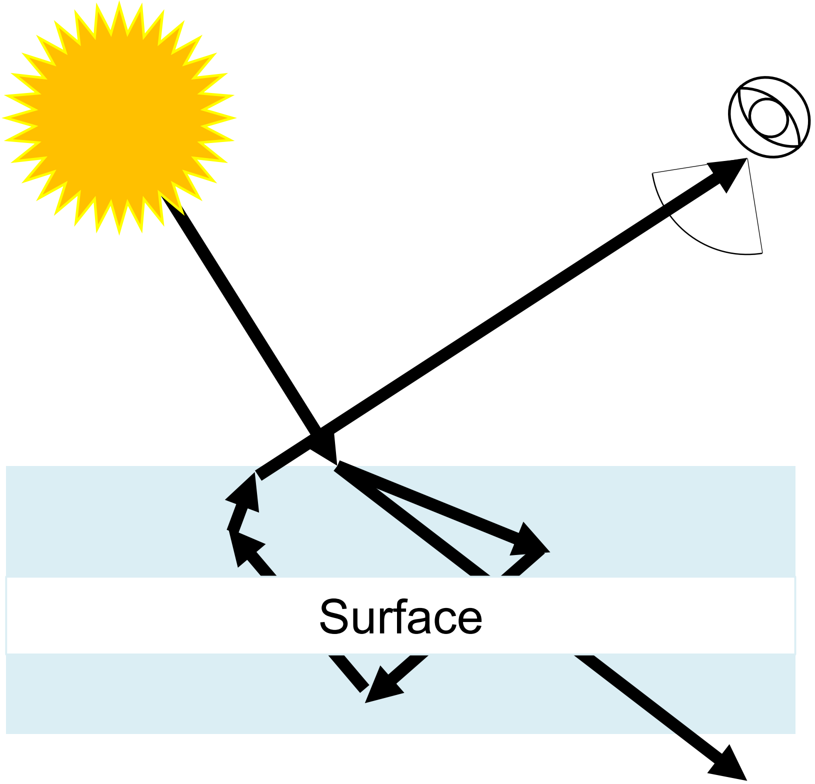


What happens when light hits a surface?

1. Absorbed

It’s absorbed and converted into some other form of energy (e.g., a black shirt getting hot in the sun)

Light and Surfaces

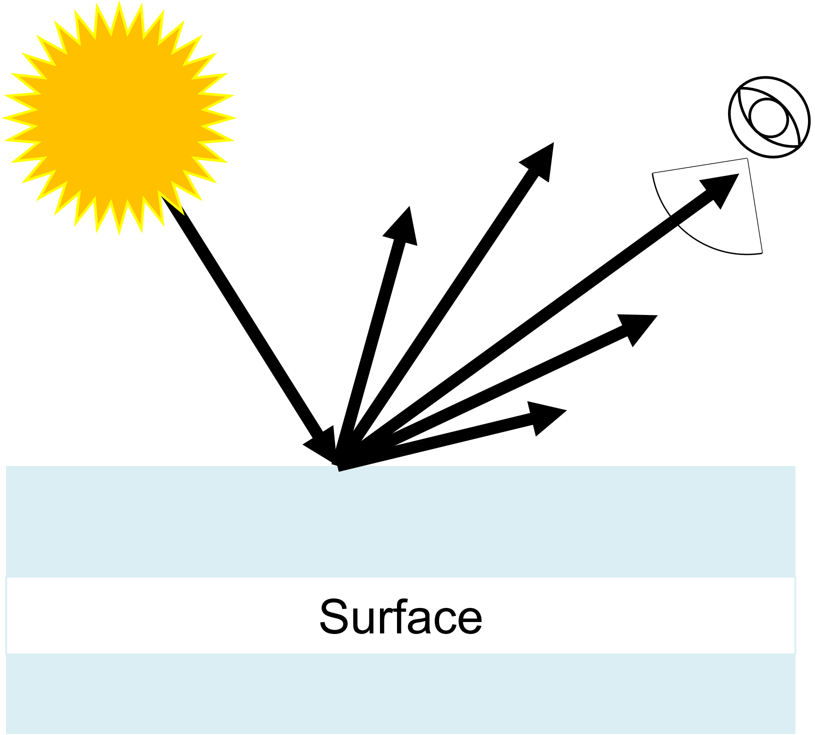


What happens when light hits a surface?

2. Transmitted

Possibly bouncing around before going through or out (e.g. lenses bend and go through, milk bounces around)

Light and Surfaces

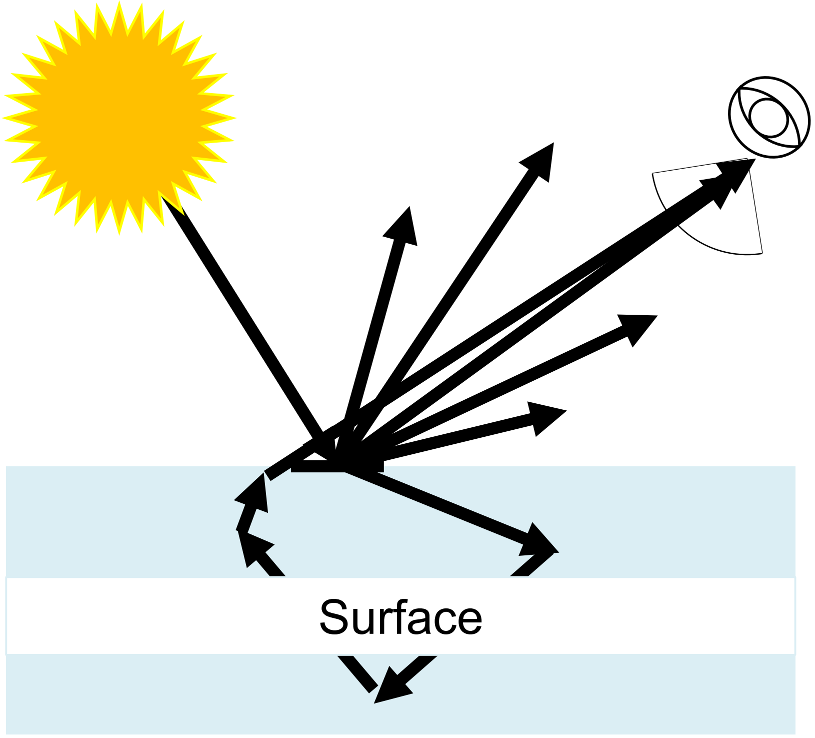


What happens when light hits a surface?

3. Reflected

It’s reflected back, in one or more directions with varying amounts (e.g., mirror, or a white surface)

Light and Surfaces

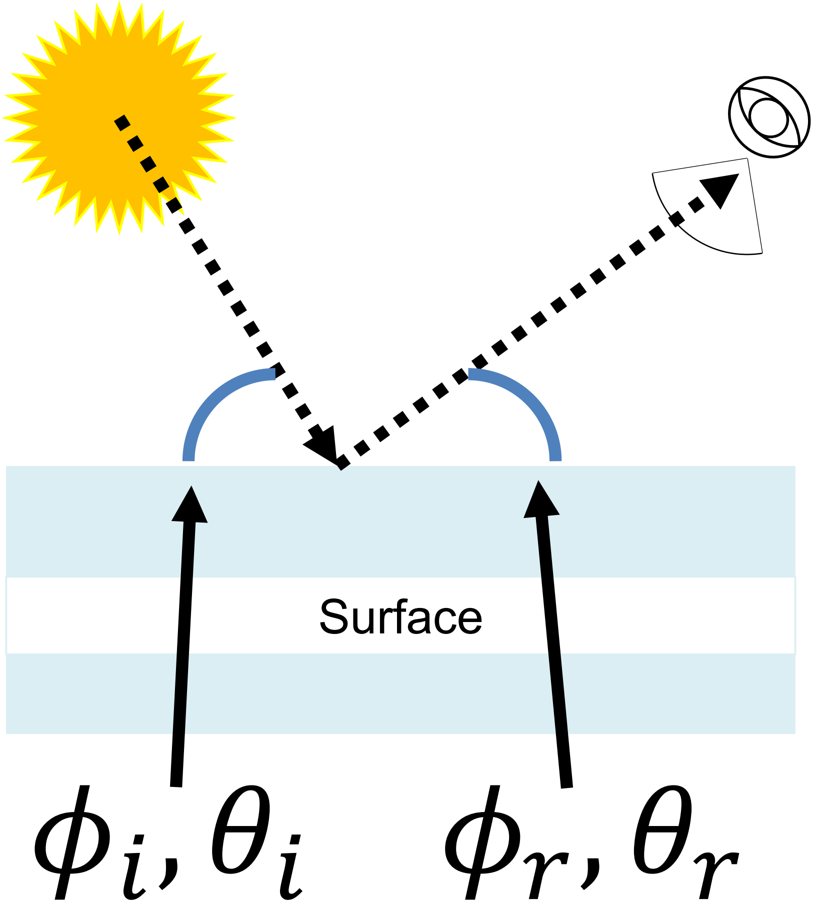


What happens when light hits a surface?

4. Everything

All of the above! Real surfaces often have combinations of all of these options.

Modeling Light and Surfaces



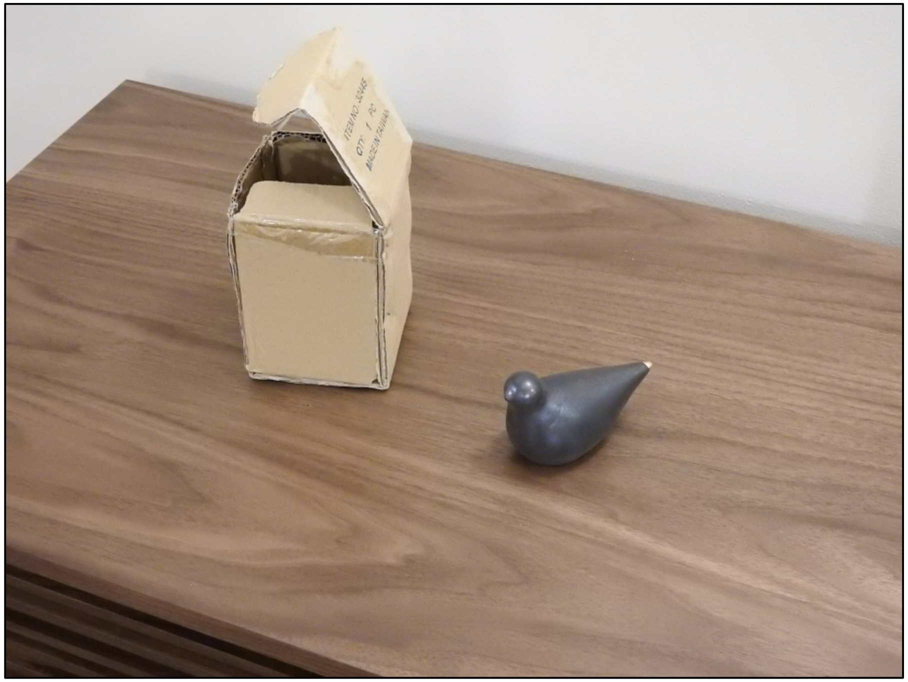
Opaque Reflections

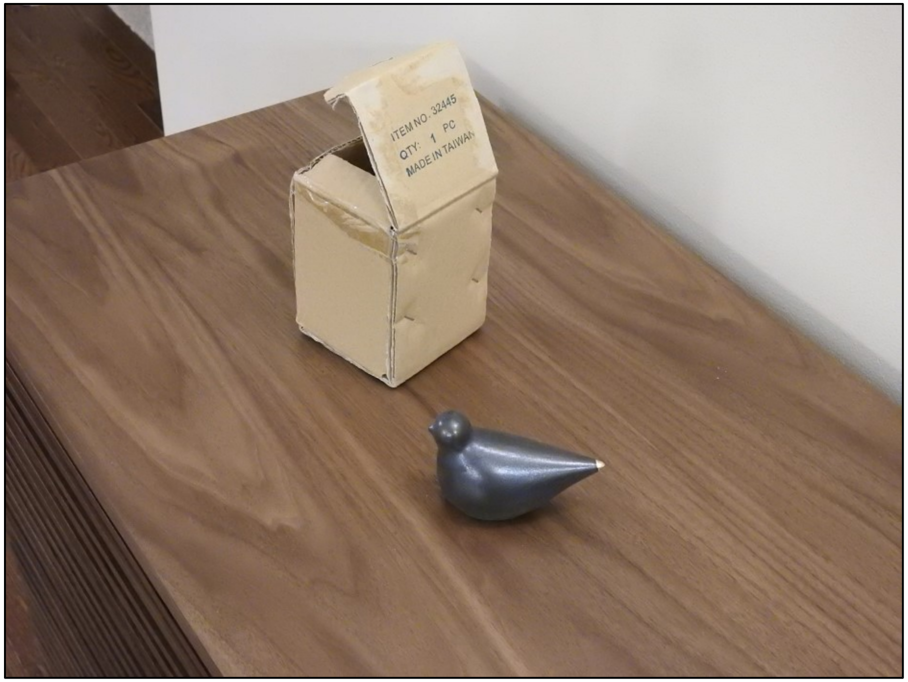
Bi-directional reflectance function: % reflected given *incident angle* to light *reflected angle* to the viewer.

Note: have not specified form of function.

Specular and Diffuse Reflection

Same lighting, as close as possible camera settings, but different **location**

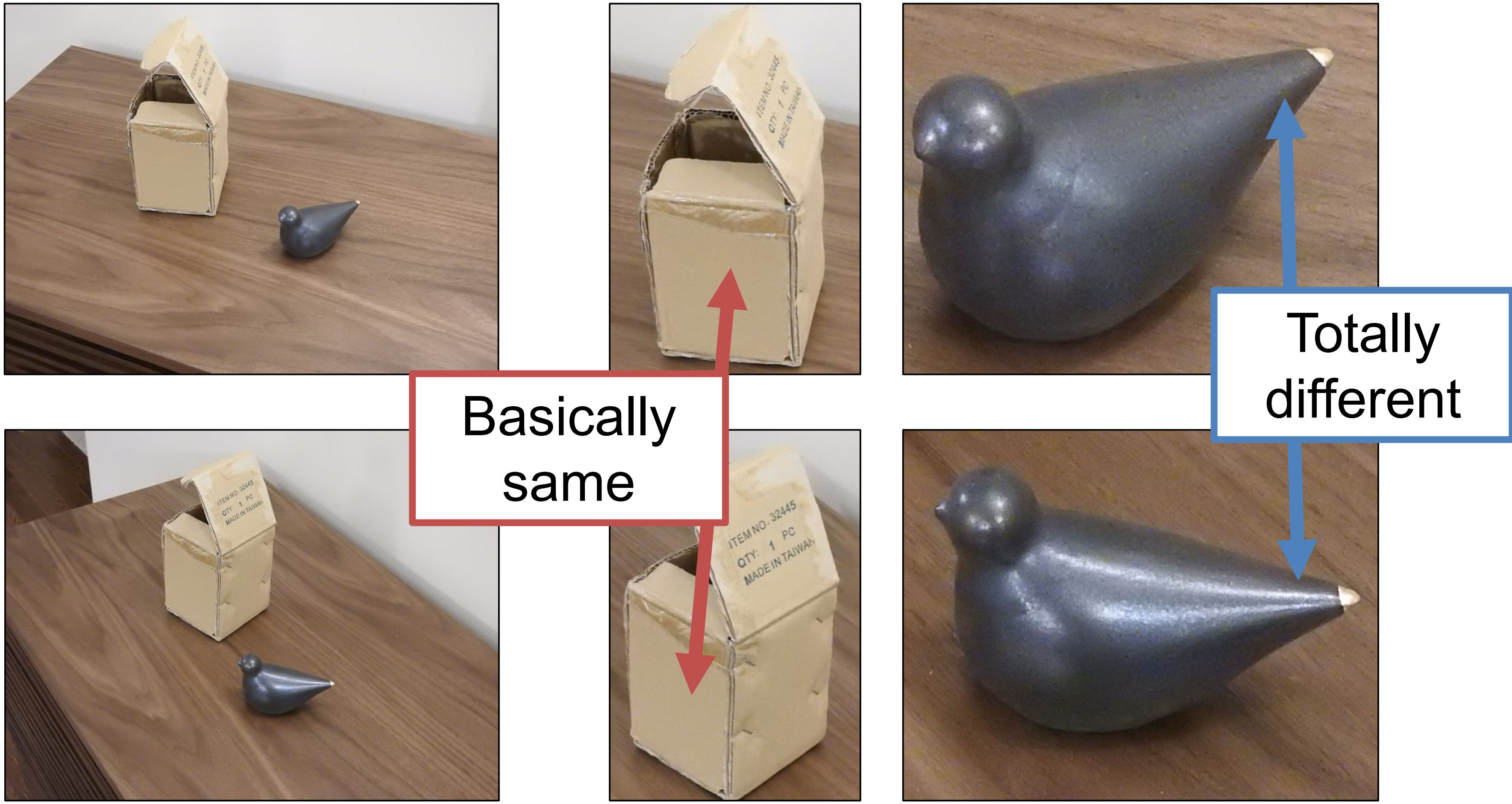




Specular and Diffuse Reflection

Diffuse

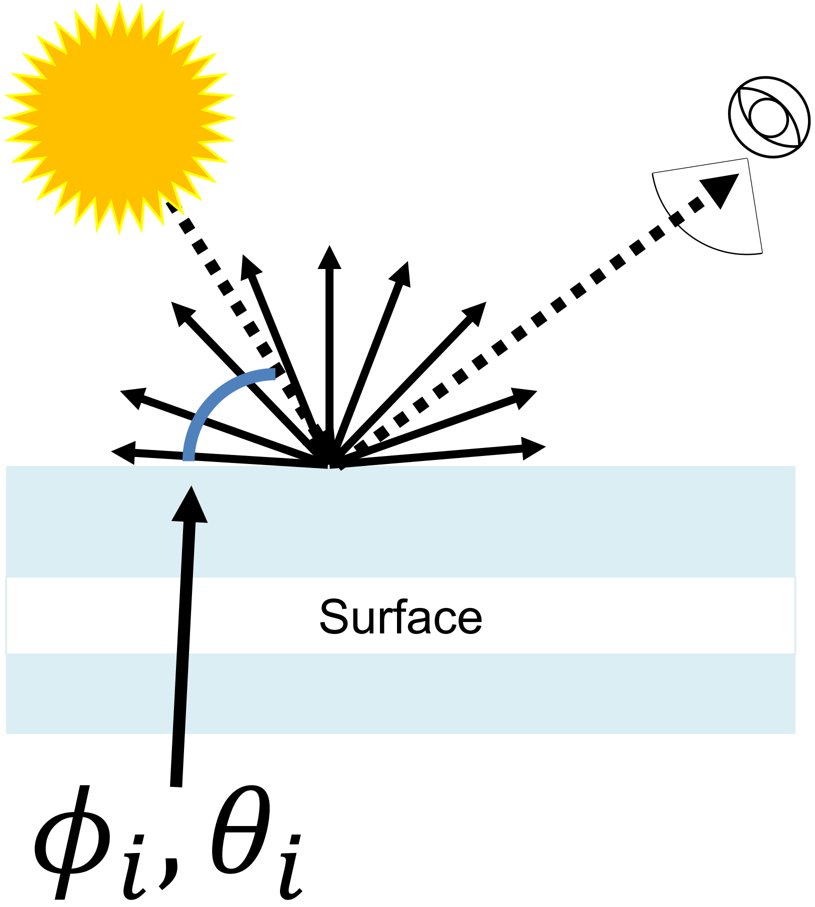
Specular



Basically same

Totally different

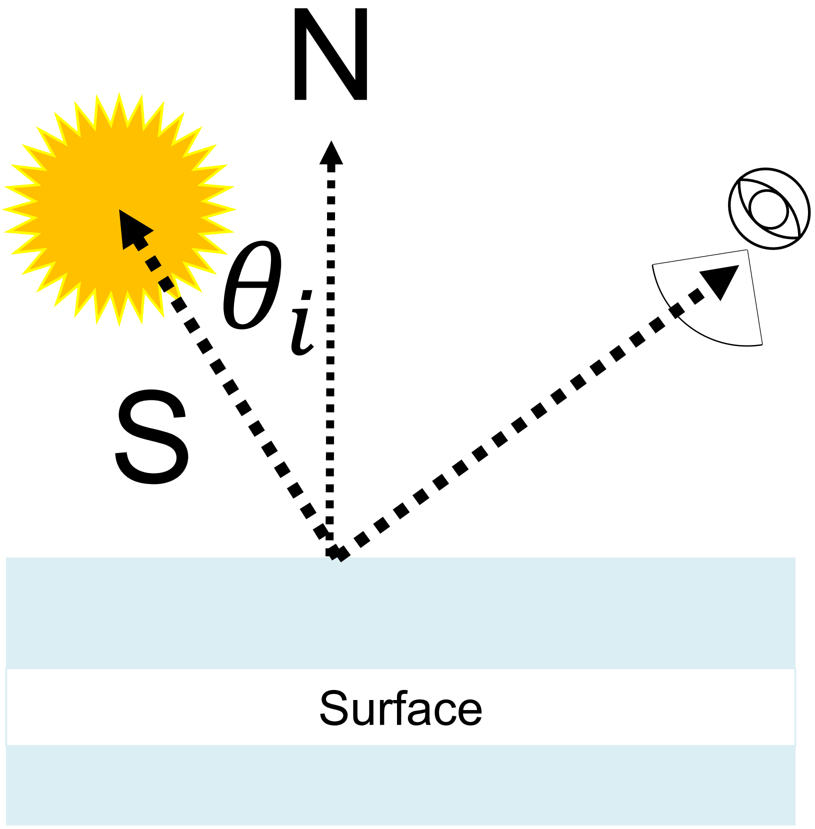
Diffuse Reflection



Lambertian Surface

Light depends **only** on orientation of surface ϕ_i, θ_i to light. Result of random small facets. Looks identical at all views.

Diffuse Reflection



Lambert's Law

N: surface normal

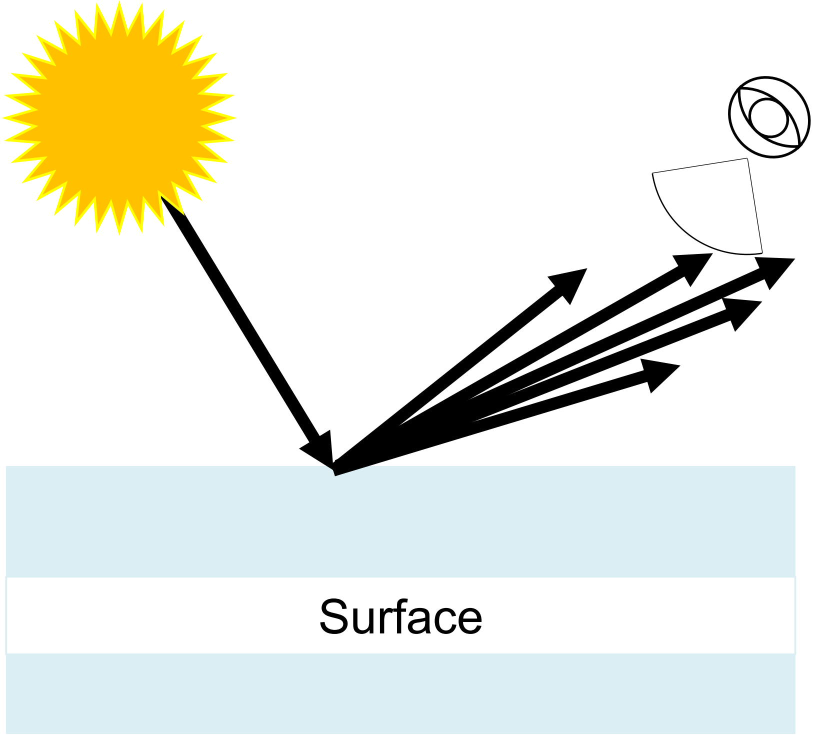
S: source direction **and** strength

ρ: how much is reflected

$$B = \rho \boldsymbol{N} \cdot \boldsymbol{S}$$

$$B = \rho ||\boldsymbol{S}|| \cos(\theta)$$

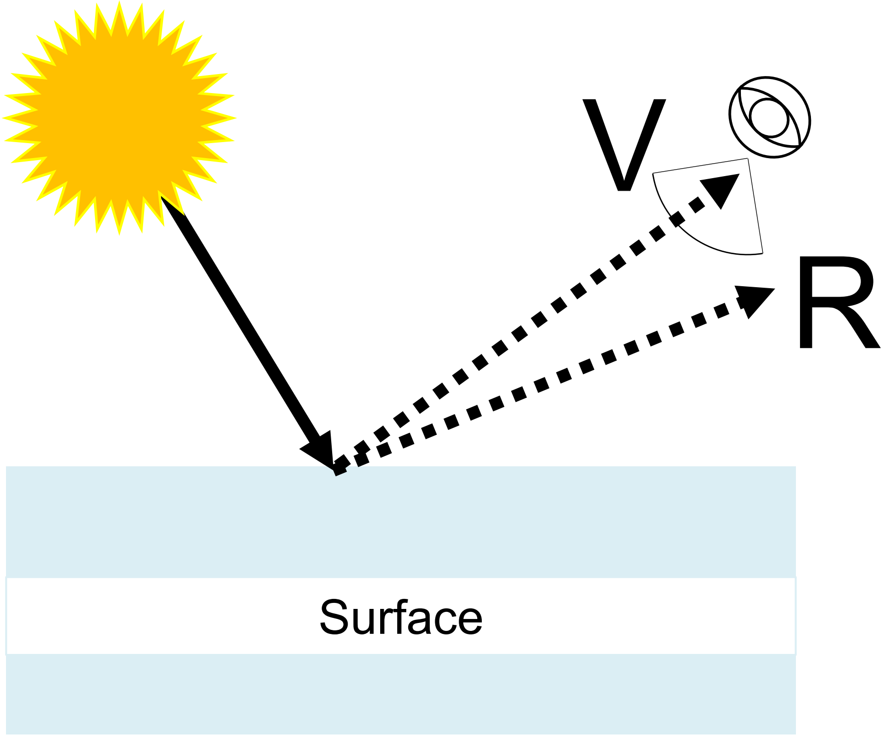
Specular Reflection



Specular Surface

Light reflected like a mirror, but spreads out in a “lobe” around the reflection ray

Specular Reflection



Phong Model

V: angle to viewer

R: reflection ray

α: shininess constant

$$B = (V^T R)^\alpha$$

BRDFs can be incredibly complicated…

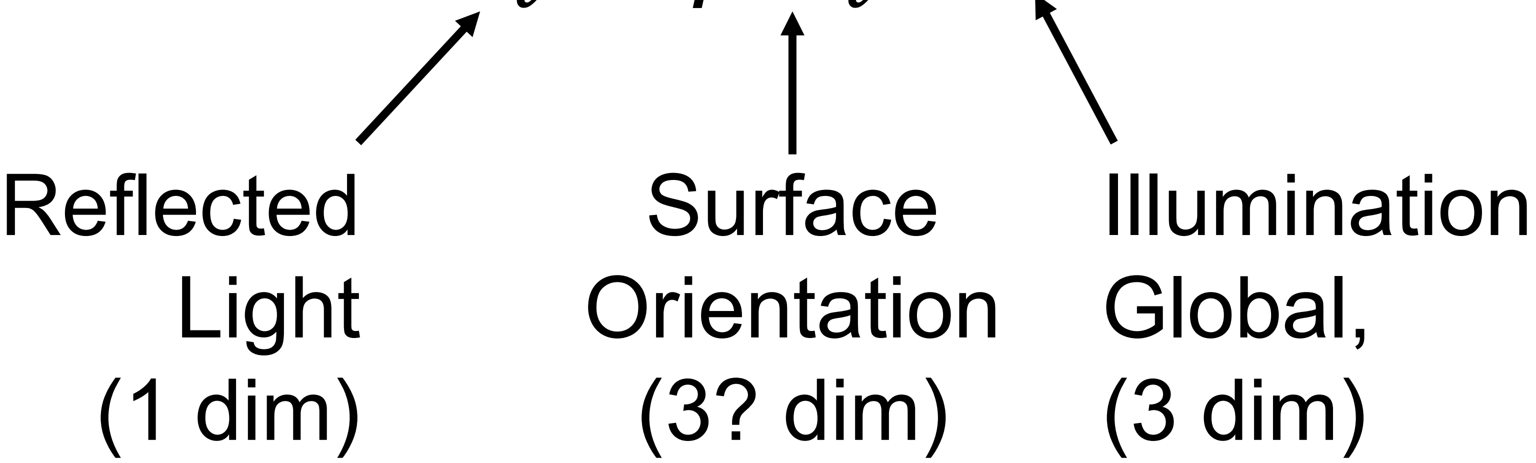
Slide Credit: L. Lazebnik

What Can This Be Used For

Shape from Shading

Lambert’s Law: for every pixel i

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$



Reflected Light (1 dim)

Surface Orientation (3? dim)

Illumination Global, (3 dim)

Given: illumination and light, recover normals

Potential problems?

Shape From Shading

$$B_i = \rho \boldsymbol{N}_i \cdot \boldsymbol{S}$$

1D, **fixed**

actually 2D **unknown**

3D, **fixed**

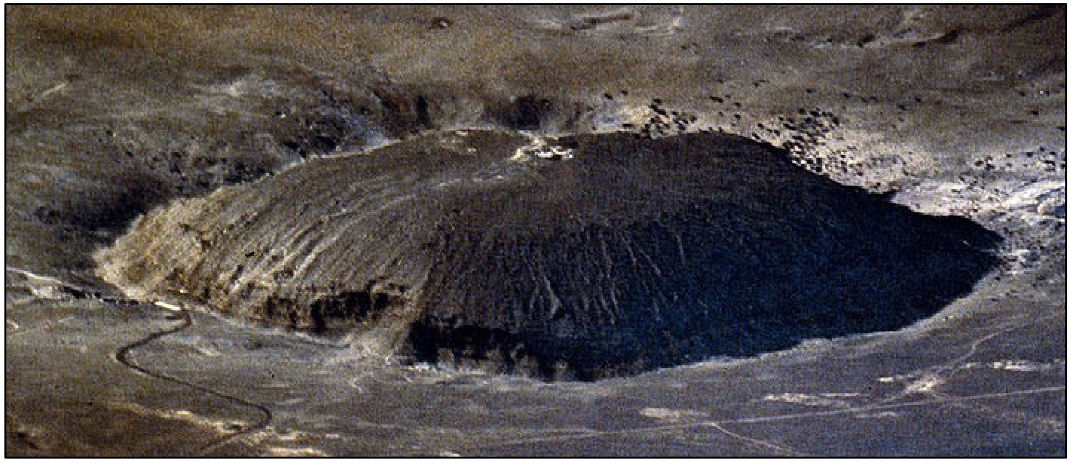
- System of equations that’s underdetermined (N equations, 2N unknowns, N+3 known)
- **Solution**: Add more equations that enforce smoothness or finding a single surface.

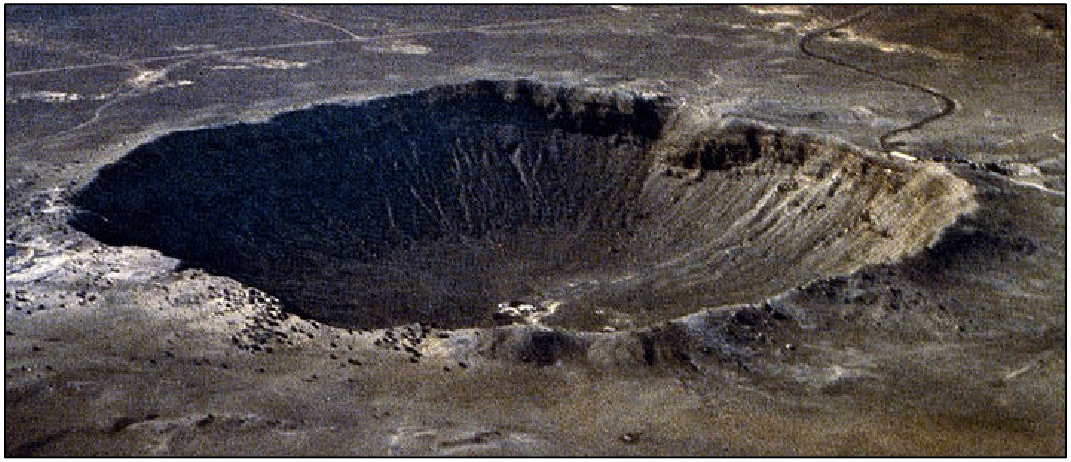
Realistic Shape From Shading

$$\underset{\substack{\nearrow \\ \text{1D, }\textbf{fixed}}}{B_i} = \rho \underset{\substack{\uparrow \\ \text{2D} \\ \textbf{unknown}}}{\boldsymbol{N}_i} \cdot \underset{\substack{\nwarrow \\ \text{3D, }\textbf{unknown}}}{\boldsymbol{S}}$$

- System of equations that’s underdetermined (N equations, 2N+3 unknowns)
- **Solution**: need prior beliefs to disambiguate.

Ambiguity





Ambiguity

Humans assume light from above (and the blueness also tells you distance)

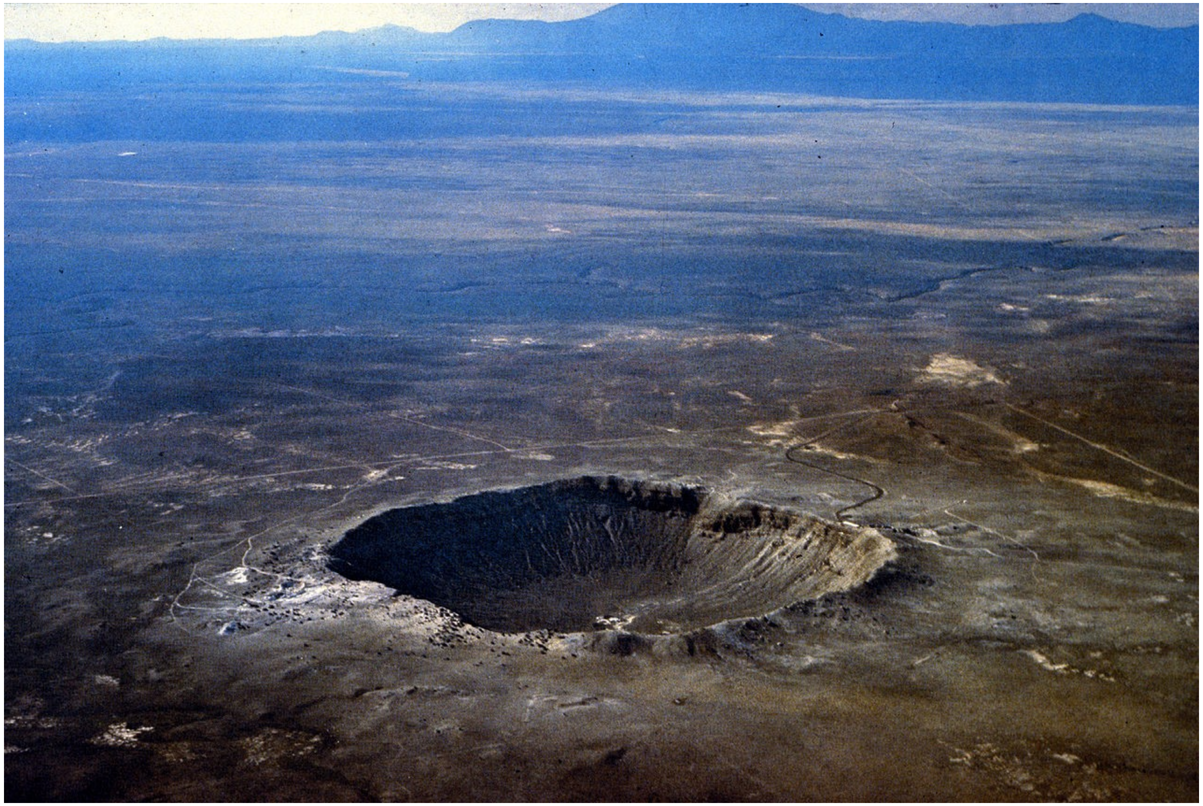


Photo Credit: https://en.wikipedia.org/wiki/Meteor_Crater

Shape from Shading in Practive

https://www.youtube.com/watch?v=4GiLAOtjHNo